

MODELING AND DETECTION OF LIMIT CYCLE OSCILLATIONS IN THIN-  
WING AIRCRAFT USING ADAPTABLE LINEAR MODELS

By

MICHAEL RAY JOHNSON

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2003

## ACKNOWLEDGMENTS

I would like to acknowledge the contributions of several individuals in my pursuit of education. First and foremost I acknowledge my father, Dr. M. Ray Johnson, who taught me that the primary purpose of a formal education is to learn how to learn, that learning is a process unique to each individual, and that learning is a life-long endeavor that more often than not doesn't involve a classroom. His example as a father, a student, a teacher and a friend has been a guide in my life.

Next I acknowledge Professor Jose Principe who, by means of his talent as a teacher and advisor (and perhaps a touch of the supernatural) understands my learning process better than I do. He always seems one step ahead of me. For his guidance, patience and friendship I am very grateful.

I would be remiss if I didn't acknowledge the support of the Air Force SEEK EAGLE Office. Dr. Bill Dyess encouraged and supported my efforts and made it possible for me to concentrate full time on this research. Dr. Chuck Denegri, my friend and mentor, was a ready and willing sounding board, without whose help this work would not have been possible.

Finally acknowledgments to my wife and kids. Words cannot describe the support and encouragement they gave. Suffice it to say that the word 'quit' was never tolerated, a closed door—an all-too-often occurrence—was always respected, and hugs were well-timed and readily available. Every guy should be so lucky.

## TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS .....	ii
LIST OF TABLES .....	vi
LIST OF FIGURES .....	vii
CHAPTER	
1 INTRODUCTION .....	1
Problem Description and Proposed Solution .....	3
Motivation .....	9
Dissertation Organization .....	10
2 BACKGROUND .....	12
The Flutter Phenomenon .....	12
Linear Flutter Analysis .....	14
A Brief Flight Test Overview .....	21
Flight Test Data Sets and Configuration Linear Flutter Analysis .....	23
Aircraft Accelerometer Locations .....	23
Mission Particulars .....	24
Linear Analysis of Flight Test Configuration .....	25
Flight Test Signals .....	27
Test Line 17 Test Signal .....	27
Test Line 3 Test Signal .....	31
Alternative Approaches .....	36
Engineering Approaches .....	36
Volterra series .....	36
Nonlinear empirical methods .....	37
NASA solutions .....	38
Similar ANN work .....	38
Physical Modeling Approaches .....	39
Computational fluid dynamics .....	39
Local linear models .....	42
3 MODELING APPROACH .....	44
LLM Approach: Quasi-Fixed IIR/Adaptive FIR Modular Approximation of an IIR System .....	44

Model Implementation.....	49
Kuo's Two-PE Oscillator as a Foundation .....	49
Spring-Mass Linear Oscillator .....	51
Conversion to Single-Weight Adaptation.....	53
Network Transfer Function Analysis.....	56
Linear Damping.....	60
Local Linear Models.....	61
Generalized Likelihood Ratio Test Basis for Detecting Signal Changes.....	63
A Proposed Modification to the GLRT .....	68
GLRT Practical Considerations.....	73
Chapter Summary: The Initial Model.....	75
<b>4 MODEL TUNING AND REFINEMENTS.....</b>	<b>77</b>
Chapter Organization.....	78
Linear Oscillator Experiments .....	79
Network Training .....	79
Frequency Training.....	83
Phase and Gain Training.....	86
Sample Frequency Training Results.....	90
Dual Sinusoid Phase and Gain Training Example .....	91
Dual Sinusoid Phase and Gain Training Example Results .....	93
Post Analysis Oscillator Design Modifications .....	98
Linear Oscillator Summary .....	99
Modified Generalized Likelihood Ratio Test for Detecting Signal Changes .....	100
Preliminary Experiments .....	102
Experiment 1 .....	104
Experiment 2 .....	109
Experiment 3 .....	111
Experiment 4 .....	113
Network Parametric Evaluation .....	116
Reduced-Model Simplified GLRT Experiment Conclusions .....	119
Fixed-Length Flight Test Signal Training Experiments.....	121
Preliminary Work: Constant Mach, Altitude and Normal Acceleration LCO	
Evaluation (Test 8c).....	121
Evaluation of the Practical Value of MSE for Training .....	125
Increased-Time Experiment Results.....	126
Consecutive Fixed Length Experiments (Test 8c) .....	132
Consecutive Fixed Length Experiments (Test Line 17).....	145
Chapter Summary.....	152
<b>5 FLIGHT TEST EXPERIMENTS .....</b>	<b>155</b>
Test Line 17 LLM Experiments.....	157
Parametric Predictions.....	160
Reduced-Order Networks .....	163
Reduced-Order Model Results .....	173
Reduced-Mode Results.....	175



Linearly-Fit LLM Parameter Results .....	177
Test Line 3 LLM Experiments .....	179
Third-Order Parametric Fit Prediction Results .....	181
Parametric Prediction Results .....	187
Reduced-Mode Results .....	187
Frequency Adaptation Analysis .....	189
Model Generalization Evaluation .....	192
Chapter Summary .....	194
<b>6 DISCUSSION AND CONCLUSIONS .....</b>	<b>197</b>
Modeling Conclusions .....	199
Linear Oscillator .....	199
Fixed-length Signal Modeling .....	200
Reduced-Model Simplified GLRT .....	201
Flight Test Data Experiments .....	202
Further analyses .....	206
Confirmation of existing methods .....	210
LIST OF REFERENCES .....	212
BIOGRAPHICAL SKETCH .....	217

## LIST OF TABLES

<u>Table</u>	<u>page</u>
2-1 Flight 8951 free vibration analysis frequencies by mode. ....	26
4-1 Oscillator trainability within 1% desired frequency.....	85
4-2 Oscillator trainability within 1% desired frequency, plastic input weight. ....	86
4-3 Dual sinusoidal frequency phase/gain test phase values. ....	93
4-4 Experiment 1 desired and ANN models. ....	106
4-5 Experiment 1 Results.....	108
4-6 Initial Modal Frequencies (free vibration values).....	122
4-7 Initial Modal Amplitudes (flutter solution results). ....	132
5-1 Test line 17 modes 1 to 8 adapted frequencies (Hz) .....	190
5-2 Test line 17 modes 9 to 16 adapted frequencies (Hz) .....	191
5-3 Test line 3 adapted frequencies (Hz).....	192

## LIST OF FIGURES

<u>Figure</u>	<u>page</u>
1-1 LLM conceptual diagram.....	6
2-1 DMM points on a wing.....	20
2-2 Aircraft configuration and sensor locations.....	24
2-3 Linear flutter analysis of test configuration, Damping.....	26
2-4 Linear flutter analysis of test configuration, Frequency.....	26
2-5 KCAS, Mach, and altitude relationship.....	27
2-6 Test line 17 sensor traces: altitude, Mach and normal acceleration.....	30
2-7 Test line 17 forward wing tip signal and power spectral density.....	30
2-8 Test line 17 aft wing tip signal and power spectral density.....	31
2-9 Test line 3 flight test states and signals. ....	32
2-10 Test 8 forward wing-tip accelerometer.....	33
2-11 Test line 3 analysis points, flight test states and signals.....	35
2-12 Test line 3 PSD, forward and aft sensors.....	35
3-1 Conceptual diagram of oscillator driven FIR filter system. ....	48
3-2 Kuo two-PE oscillator network.....	50
3-3 Original four-weight spring-mass oscillator network. ....	52
3-4 Spring-mass oscillator output. ....	53
3-5 Single weight oscillator network.....	56
3-6 Single weight spring-mass oscillator output.....	56
3-7 Impulse response of total system and two part system.....	57
3-8 Multi-modal impulse response.....	58

3-9 Three model GLRT. ....	64
4-1 Single-weight oscillator network. ....	81
4-2 Single-weight oscillator architecture.....	81
4-3 Initial signal trace of linear oscillator and desired response.....	82
4-4 Trained signal trace of linear oscillator and desired response. ....	82
4-5 Oscillator frequency weight curves. ....	84
4-6 Complete single frequency oscillator architecture.....	87
4-7 Gradient surface for 7 Hz initial frequency. ....	88
4-8 Gradient surface for 14 Hz initial frequency. ....	89
4-9 Freq. training test, two cycles. Trained input weight, 7 to 9 Hz. ....	91
4-10 Freq. training test, one-second. Trained input weight, 14.0 to 14.5 Hz. ....	92
4-11 Complete double-frequency oscillator architecture. ....	93
4-12 Dual frequency output with trained input weights. ....	94
4-13 Dual frequency with trained input weights training results.....	95
4-14 Dual frequency output with fixed input weights.....	97
4-15 Dual Frequency output with no trained input training results. ....	97
4-16 Modified saturated linear activation function. ....	98
4-17 Experiment 1 test signal.....	105
4-18 Experiment 1 L-values.....	107
4-19 Experiment 2 test signal comparison.....	110
4-20 Experiment 2 L-values.....	111
4-21 Experiment 3 test signal comparison.....	112
4-22 Experiment 3 L-values.....	113
4-23 Experiment 4 test signal comparison.....	115
4-24 Experiment 4 L-values.....	115
4-25 Experiment 3 network parameter comparison. ....	117

4-26 Experiment 4 network parameter comparison. ....	118
4-27 Complete 16-mode LLM architecture. ....	122
4-28 Test 8c Spectrogram. ....	125
4-29 One-eight second signal training test. ....	127
4-30 One-quarter second signal training test. ....	127
4-31 One-half second signal training test. ....	128
4-32 One second signal training test. ....	128
4-33 Four second signal training test. ....	129
4-34 Five second signal training test (failure). ....	129
4-35 Test 8c quarter-second burst training results. ....	137
4-36 Test 8c quarter-second burst network parameters. ....	138
4-37 Test 8c one-second burst training results. ....	140
4-38 Test 8c one-second burst network parameters. ....	141
4-39 Test 8c one-second prediction plots. ....	142
4-40 Test 8c four-second burst training results. ....	143
4-41 Test 8c four-second burst network parameters. ....	144
4-42 Test line 17 forward wing tip accelerometer signal and PSD. ....	146
4-43 Test line 17 original architecture training results. ....	147
4-44 Test line 17 original architecture network parameters. ....	148
4-45 Complete two-delay oscillator architecture with output bias. ....	149
4-46 Test line 17 two-delay, biased architecture training results. ....	150
4-47 Test line 17 two-delay, biased architecture network parameters. ....	151
5-1 Complete 2-delay, 16-mode LLM architecture. ....	156
5-2 Results of Test line 17 auto-switched LLM data synthesis. ....	158
5-3 PSD for test line 17 auto-switched LLM data synthesis. ....	159
5-4 L-values for test line 17 auto-switched LLM data synthesis. ....	159

5-5 Test line 17 zero-order FIR weight (mode 1) vs. Mach. ....	161
5-6 Test line 17 zero-order vs. second-order FIR weights (mode 1). ....	161
5-7 Test line 17 first order FIR weight (mode 1) vs. Mach. ....	162
5-8 Test line 17 LLM bias vs. Mach. ....	163
5-9 Test line 17 reduced-order LLM FIR modal vector angles by segments. ....	167
5-10 Test line 17 reduced-order FIR weight vector angles vs. Mach. ....	170
5-11 Test line 17 reduced-order FIR zero-order vs. first-order weights. ....	171
5-12 Test line 17 reduced-order FIR weight vectors.....	172
5-13 Test line 17 synthesis using reduced-order FIR filters. ....	174
5-14 PSD of test line 17 synthesis using reduced-order FIR filters. ....	174
5-15 Data synthesis of test line 17 using LLMs for modes 1,2, and 7. ....	175
5-16 PSD comparison of test line 17 using LLMs for modes 1,2, and 7. ....	176
5-17 Data synthesis of test line 17 using linear-approximated LLMs.....	177
5-18 PSD comparison of data synthesis using linear-approximated LLMs. ....	179
5-19 Test line 3 reduced order results. ....	181
5-20 Test line 3 reduced-order LLM FIR modal vector angles by segments. ....	182
5-21 Test line 3 reduced-order FIR weight vector angles vs. g-loading. ....	184
5-22 Test line 3 reduced-order FIR first-order vs. second-order weights. ....	185
5-23 Test line 3 reduced-order FIR weight vectors.....	186
5-24 Data synthesis of Test line 3 using Third order approximated LLMs.....	188
5-25 Data synthesis of Test line 17 using LLMs for modes 1,2, and 7.....	189
5-26 Data synthesis of test line 3 segment 1, LLM for Test line 17 Model 1. ....	193

Abstract of Dissertation Presented to the Graduate School  
of the University of Florida in Partial Fulfillment of the  
Requirements for the Degree of Doctor of Philosophy

MODELING AND DETECTION OF LIMIT CYCLE OSCILLATIONS IN THIN-  
WING AIRCRAFT USING ADAPTABLE LINEAR MODELS

By

Michael Ray Johnson

August 2003

Chair: Jose C. Principe

Major Department: Electrical and Computer Engineering

A method for modeling the flutter response of a thin winged aircraft is presented in this dissertation. Flutter is a behavior of a dynamical system subjected to aeroelastic forces. Adaptive systems can model the flutter response, but require complicated Auto Regressive, Moving Average (ARMA) architectures to do so.

To simplify the task, this work proposes a methodology to separate the AR and MA components into a hybrid physical model consisting of a single-input multi-output (SIMO) adaptive AR model feeding into an adaptive multi-input single-output (MISO) adaptive MA model providing the necessary signal changes due to forces encountered by the wing during flight. Precise AR pole placement is accomplished by adaptive oscillators set at the modal frequencies identified by a free vibration analysis of the wing, thus representing a physical model of the structure.

Zero placement is then accomplished by feeding the oscillator outputs into a bank of adaptive linear finite impulse response (FIR) filters that are adapted to the flight test

sensor data. Together these modules form a hybrid model for the wing flutter under flight conditions.

In order to cope with the nonstationary nature of the task, the output of the models is segmented using statistical methods based the Generalized Likelihood Ratio Test (GLRT) using maximum likelihood estimates of the error distributions. The overall sequence of models provides a complete synthesis of the nonlinear response recorded by the accelerometer as flight conditions change. The sequence of network parameters is analyzed for frequency, phase and gain of each mode.

The paradigm is used with two portions of flight test data. It performs very well against data taken at increasing Mach numbers in level flight. Network parameters are shown to correlate very well with the Mach numbers. The paradigm shows promising results when used against more complicated flight scenario with changes in normal accelerations.



## CHAPTER 1 INTRODUCTION

One of the most interesting problems associated with flight, a problem that has plagued engineers since powered flight began, has been the interaction of aerodynamic, inertial, and structural forces that result in a structural response known as flutter. To further complicate the issue, modern engineering has noted and studied a subclass of flutter dealing with the phenomenon of Limit Cycle Oscillations (LCO). These oscillations are a persistent problem on modern fighter aircraft with thin, flexible wings. Since safety and cost are paramount in the modern world of aviation, new ways to describe, model, and simulate these responses are always being sought.

One advantage that modern flight test organizations now have is the large amount of data that can be obtained from onboard instrumentation systems used during a flight test. The signals from these sensors are used by aeroelasticians to evaluate a particular aircraft response, and with the increase in data availability comes an increase in the options available for flutter and LCO analysis. This work will address the study of LCO not from a traditional aeroelastic perspective, but instead as a challenge to be answered using modern signal processing techniques. The data contained in the instrumentation signals can provide fresh insight into the structural behavior of the aircraft. Using a quasi-linear Artificial Neural Network (ANN) architecture this work goes beyond evaluating the LCO response by simple observation of envelop characteristics to a process that results in information giving a clearer understanding of the modal construct of the signal and the response of the structure itself.

The particular variety of LCO this work will focus on is that observed during transonic flight on the wing. It is generally accepted that LCO arises from the interaction of structural and aerodynamic forces acting on the affected aircraft component,<sup>1</sup> and that this interaction is nonlinear in nature. Much research and debate has been devoted to determining the contribution of each force, as well as their interaction. As defined, classical flutter is a phenomenon such that once oscillations initiate they catastrophically diverge. Limit cycle oscillations is a phenomenon described Denegri<sup>2</sup> as a variety of classical flutter, but differs in its tendency toward limited amplitude oscillations rather than diverging oscillations. As such, this work will treat flutter and LCO as the same phenomenon. The same background, theory, fundamental assumptions, model structure and paradigm will be used for both.

In recent years research in flutter analysis has taken two broad paths: empirical modeling, the more traditional engineering approach to problem solving; and physical modeling, the most notable example of which might be the advances in Computational Fluid Dynamic (CFD) simulations. Empirical modeling seeks to mathematically describe the phenomenon and provide an analytical method for classifying a system's response. Physical modeling may also include empirical methods, but often includes methods that model the structural behavior of a system based on physical traits or characteristics.

There are, of course, many areas of research and development interested in furthering the efforts to more effectively model the flutter phenomenon. One such area, compatibility certification, focuses on using models, simulations, and flight testing to certify an aircraft for safety of flight when different stores are hung in, or on its structure. The compatibility certification process uses classical linear flutter analysis methods and assumptions as the current standard for determining flutter sensitive loads and flight

conditions. These linear methods are approximations that have been considered reliable, and can be categorized as empirical modeling. Theoretical linear flutter analysis can adequately identify flutter or LCO sensitive store configurations and the frequencies<sup>3</sup> at which the instabilities may occur. Additionally, correlation between the flight test response and the modal composition of the analytical flutter mechanism is evident<sup>2</sup>. But because the response is nonlinear, classical linear flutter analysis techniques fail to predict the onset or severity of the LCO.<sup>4</sup>

In this work the challenge of LCO modeling is approached as a hybrid of a physical modeling problem in that a physical model of the wing modes is modified by adaptation. More specifically, the problem is approached from an adaptive signal processing perspective by exploiting the ability of a series of Local Linear Models (LLMs) to model a complicated time-dependent, nonlinear data stream. Training adaptive systems, most recently ANNs, using the measured time series of a dynamical system is now common.<sup>5</sup> The usefulness of adaptive systems in general has been shown in system identification and prediction of nonlinear signals similar to those from the on-board accelerometers used in flutter and LCO flight testing.<sup>6</sup> They have also been effective in predicting the amplitude and frequency of flutter and LCO test results.<sup>7-9</sup> The use of LLMs has proven effective in modeling aircraft dynamics<sup>10</sup> and wind tunnel characteristics;<sup>11</sup> works similar to this study that lend encouragement to the applicability of LLMs for this work.

### **Problem Description and Proposed Solution**

A problem that has inhibited the advancement of empirical methods for accurately determining the onset and nature of responses in flight such as LCO is that there is a large analytical gap between the current empirical and modeling tools and their ability to match actual in-flight responses. Most techniques are linear and time invariant, while the actual

response is highly nonlinear and time varying. Flight test data is difficult to analyze from anything more than a macro point of view, so empirical model refinement has been a slow process. Although many advances have been made in the number and types of sensors available to measure the response of an aircraft, flight test data for the measurement and analysis of aircraft flutter and LCO is still expensive to gather and limited to data from a small number of accelerometers located on the aircraft. Further, accelerometers only provide the total response of the wing at the sensor location, so it is not possible to completely define the modal components of the response. This limits the type of analysis that can be accomplished. There is little on-line prediction or post-flight analysis that can be done.

Physical models are able to replace or enhance analysis flight test data for flutter analysis. The use of CFD, for instance, has become more practical in recent years as computer speed and efficiency increases. But CFD is still very computationally intensive. A CFD simulation of one aircraft configuration at one flight condition can take several hours to run. Since a single flight test for a given aircraft configuration can contain nearly 100 test conditions it is easy to see that CFD is still far from being a practical alternative.

Physical models may be able to provide insight into the modal construct of a response and provide information towards improved empirical models if designed properly. Flutter is a behavior of a dynamical system subjected to aeroelastic forces. Linear flutter analysis suggests that the flutter response can be approximated as a dynamical process whose behavior can be modeled as a sum of sinusoids for a given flight condition. If this is correct, it can be inferred that a flutter response can, under some conditions, be considered a linear, time invariant process for a short period of time.

Linear flutter analysis provides a very sound structural model since it is based on a dynamic mass model (DMM) of the wing, which is itself a derivative of a more complex finite element model. Assuming that this suggestion is accurate, these concepts can be exploited to help describe and design a hybrid physical model of the system. An adaptive system can be constructed in such a way as to provide as an output the sum of sinusoids suggested by linear flutter analysis, trained to track a burst of accelerometer data measuring the response of a fixed, steady state flight condition. It follows that if constructed properly a decomposition of the trained system would be an accurate representation of the modal construct of the actual signal. The system must be continuously adapted, since linear methods do not completely describe the time varying aerodynamic forces encountered.

Using these hypotheses as a guide, and assuming that a response can be considered stationary and time invariant for the period of time required for adaptation, a hybrid physical Multiple Input Single Output (MISO) model for the wing accelerometer is proposed, constructed as a modular ANN providing summed, adapted oscillations to simulate a steady state window of an accelerometer's signal (Figure 1-1). A family of these models is assembled or trained so that their outputs, evaluated as a sequence, provide a complete synthesis of the nonlinear response recorded by the accelerometer as flight conditions and dynamics change. This system is a hybrid of a physical modeling process since it is based on knowledge of the physical system (the modes), but also data driven. Adaptation of the system to the flight test data provides the necessary parameterization between linear techniques and actual forces. The components of the ANN are such that parameters for each mode in the response can be evaluated separately, providing for the first time the ability to monitor the contribution of each mode to the

system response, and the changes to the modes as a function of flight dynamics. The system is illustrated in Figure 1-1.

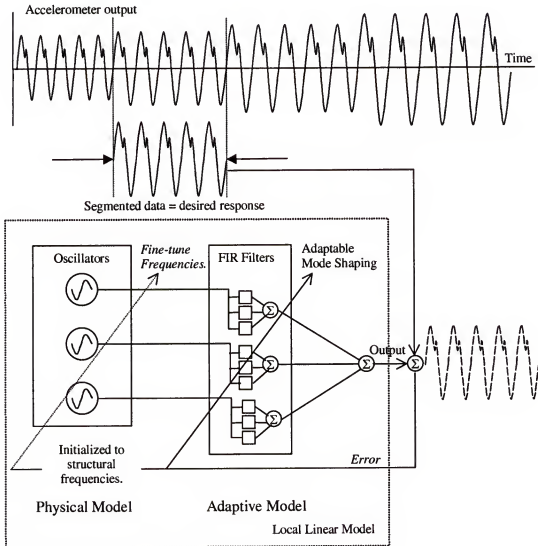


Figure 1-1 LLM conceptual diagram.

Modal oscillations are provided by a bank of linear oscillators, set at the free vibration frequencies identified by linear flutter analysis. As such, these oscillators represent a physical model of the structure since the frequencies are initialized to those identified by a free vibration analysis of the wing. The output of each oscillator is input to a bank of adaptable Finite Impulse Response (FIR) filters. There is one filter for each

mode, allowing the phase and amplitude of each frequency to be adjusted independently, matching the forced response of the structure. The filtered oscillations are then summed, providing the complete model output.

The proposed LLM consists of carefully designed adaptive signal processing components using linear and nonlinear Processing Elements (PEs) connected as ANNs. This architecture allows precise pole placement using an Auto Regressive (AR) module, and FIR zero placement with a separate Moving Average (MA) linear structure connected in series. Both modules are adaptable. The advantage of the modular architecture is that the poles can be placed with high precision prior to training based on the structural a priori information available. This reduces the adaptation of the poles to simple fine-tuning of the modal frequencies as they may drift during the response. The zeros are placed at their corresponding free vibration values as well. However, since linear flutter analysis does not provide phase information, the relative phase of each mode is not initialized. Once adapted to a burst of an accelerometer's signal, this modular architecture can be analyzed for frequency, phase and gain of each mode contained in the signal.

An extrapolation of the system is that if the LCO response is locally time-invariant, that is to say that for a given flight condition the homogeneous response of the wing is fixed, a linear model could be trained for successive, or separate portions, or windows, of that response, forming a family of LLMs. While most applications of LLMs define each model as a region in state space, the proposed method adapts models to regions in temporal space, although each model may be linked to specific states. The models are not selected for reuse as a function of state. Instead they are evaluated for applicability directly to the desired response using the Mean Squared Error (MSE) as the

criteria. The effect is that models are selected or trained as states change, providing a time-varying model to a time-varying response that could be mapped to specific states. It is possible, however, that since the system is dynamic with an Infinite Impulse Response (IIR) it is foreseeable that more than one response is measured at a specific state, resulting in more than one model mapped to the same place in state space. It is for this reason that a paradigm for temporal data segmentation rather than state modeling is proposed.

The flight test accelerometer data, which serves as the desired response for network training, is segmented when the accelerometer signal's characteristics indicate that a change in model is necessary. Changes in the signal are detected by using a modification to the Generalized Likelihood Ratio Test (GLRT), a statistical evaluation technique that monitors the errors between the models of that signal. Once a change in the signal is detected, a database containing known models is queried for a better match. If no better model is available, or if a new configuration is being evaluated, an updated model is trained.

There are alternatives to the proposed data segmentation process. The signal could simply be tracked with a continuously updated adaptive system, responding to local (in time) properties of the signal. This requires the adaptive system to have finite effective memory, controlled by learning rates in systems using LMS for training. The problem arises as to how to determine and control these factors. There is a constant trade-off between speed of convergence and misadjustment after convergence.

Segmentation<sup>12 p. 117</sup> reduces the training problem by limiting the tracking to a small, presumably stationary region of data. Segmentation can detect a change between sequential but statistically different adjacent temporal regions. Thus the approach is



global in time, but local in the model, reducing the effect of modeling errors for a given local model, and increasing the ability to globally model a complex, nonlinear process. Segmentation is therefore the better choice for the problem of LCO data synthesis.

This LLM paradigm is useful as an on-line prediction tool since model parameters can themselves be modeled or clustered as a function of flight test states and used to predict ahead in time. Parametric modeling is useful as a post-flight analytical tool as well. Parametric evaluation of the LLMs will lend insight into the sensitivity of the wing response to state variables over time.

This work is intended as a demonstration of a concept, and as such will be limited to one aircraft and store configuration. But the paradigm developed will be applicable for LCO analysis in general.

### **Motivation**

There are two significant motivations for this work. First and foremost is cost of flight testing. At the time of this writing a typical mission to test several flutter sensitive flight test conditions will cost at least \$50,000 for each aircraft/store configuration, and many times much higher. Therefore whatever efforts leading to a more accurate description or prediction of necessary flights are of prime interest.

The second motivation is to provide further insight into the modal behavior of the aircraft components during flight, supporting research into better engineering tools for flutter and LCO prediction. Linear methods are targeted by researchers for expansion to include nonlinearities in several portions of the governing equations. One work in particular, a Ph.D. dissertation by Denegri,<sup>13</sup> attempted to extract information to further characterize stiffness and damping in the wing by using limited data sets from pressure sensors and accelerometers located on a test-wing using wind tunnel data. An

assumption that only one mode was primarily responsible for the response was necessary since individual modes could not be determined from the sensor data. This assumption will no longer be required since the proposed LLM paradigm provides information for as many modes as deemed necessary to sufficiently model the signal.

### **Dissertation Organization**

This dissertation is organized into six chapters. The first chapter has served as an introduction to the problem and the proposed solution and provided some motivation for the work. Chapter 2 gives background into the flutter phenomenon, and its analysis, the process by which flight testing is performed, linear flutter analysis of the flight from which the data in this work was taken and a description of that data, as well as a review of current and related methods.

Chapter 3 presents the modeling approach. The physical model is developed and the concept of the LLM and its applicability to this work is discussed. A simple recurrent network is developed to provide sinusoidal signals as inputs to the adaptive FIR filters. Signal segmentation through the use of a simplified GLRT is discussed.

Chapter 4 is an extensive set of preliminary experiments designed to further refine the model and the paradigm used for LCO modeling. Three types of experiments are conducted. The first looks at the linear oscillator, the second evaluates the simplified GLRT for signal change detection, and the third begins the process of modeling actual flight test data for fixed lengths of time.

Chapter 5 is a presentation of results of experiments with the complete paradigm implemented on two distinctly different sets of flight test data. The first set looks at a portion of data taken during level flight with only Mach number changing. Results of these experiments are very successful and provide insight not only into the modeling

process but to the modal content and behavior of the signal. The second set of data is taken over portions of flight with changing normal acceleration. This data stretches the ability of the paradigm and while the results are not as conclusive they offer further insight into the usefulness and limitations of the paradigm.

Final discussions and conclusions are presented in Chapter 6. The results are evaluated and several follow-on topics are proposed.

## CHAPTER 2 BACKGROUND

A method for modeling flight test accelerometer data is in this dissertation. In order to more fully understand the value of this model and the nature of the data itself, it is helpful to review the flutter phenomenon and its analysis, focusing on linear flutter analysis, and the process by which flight test is performed for its evaluation. One of the goals of this chapter is to provide that review. Following that, the linear flutter analysis results for the flight from which the data for this work was taken will be presented and discussed. The data sets will be introduced and details given about the signals such as flight states, frequency content, and behavioral trends. Finally, a review of current and related methods is presented.

### **The Flutter Phenomenon**

Rodden,<sup>14</sup> gives a very good historical, conceptual and mathematical overview of flutter as an aeroelastic phenomenon as well as addressing many current methods for flutter modeling. This discussion of flutter is due in great part to his work. Flutter, a form of dynamic instability, is a self-excited vibration that is possible if a disturbance to an aeroelastic system gives rise to unsteady aerodynamic loads such that the ensuing periodic motion can be sustained. At the flutter speed a critical phasing between the motion and the loading permits extraction of an amount of energy from the air stream equal to that dissipated by internal damping during each cycle and thereby sustains a neutrally stable periodic motion. At lower speeds any disturbance will be damped, while at higher speeds disturbances will be amplified. Therefore the term flutter speed is defined as the minimum speed at which the motion will sustain neutrally stable

oscillation. The motion will, in all likelihood, diverge at speeds higher than the flutter speed. The term classical flutter is used by Rodden to denote an instability that typically, but not necessarily, arises from a critical coupling of two or more modes of motion, each of which may be stable by itself.

Flutter is certainly not limited to aeronautical systems. Two interesting structural cases are cited by Den Hartog.<sup>15</sup> The more widely known is the Tacoma Narrows bridge failure. It is thought that part of the lateral bracing under the deck had partially failed, weakening the deck in torsion. On the morning of November 7, 1940 there was sufficient wind (42 mph.) to cause a sustained torsional vibration in the deck, that ultimately resulted in destruction of the bridge. The second example is that of oscillations in submarine periscopes. Since they are long, cylindrical cantilevers moving through a dense fluid (relative to aerodynamic problems) they have been known to vibrate with such severity that a blurring of the view through the scope is seen. Compensation, either structurally or through changes in speed must be made in order to use the instrument.

There are several nonlinear aspects to flutter (the definition of which is beyond the scope of this dissertation). Structural nonlinearities may limit the amplitude of flutter so that a large amplitude motion is stable while a small amplitude motion is unstable. Aerodynamic nonlinearities may have the same effect if a large amplitude motion causes a condition of flow separation. Conversely, small amplitude motions may be stable while large amplitude motions are unstable. Rodden states that large amplitude flutter is not generally a practical consideration, and for the most part it is therefore justifiable to treat flutter as a small disturbance stability problem that can be analyzed by linear techniques. If linear techniques prove insufficient, the alternative is to use techniques that solve the equations of motion recursively such as CFD.

Flutter analysis requires the determination of both the flutter speed and frequency. The analysis is, therefore, a double characteristic value problem, contrasted with vibration analysis that has only a single characteristic value; the frequency. The methods of analysis are dictated by the limited knowledge of unsteady aerodynamic forces. As knowledge is gained, more and more exact solutions can be found, hence the vast amount of ongoing research dedicated to the field. Presently, more extensive theoretical solutions are available for the case of steady state harmonic motion because of the relative simplicity of the mathematical formulation, so it is convenient to assume harmonic motion and use its methods for finding flutter speed and frequency. This is the premise of classical linear flutter analysis.

### Linear Flutter Analysis

Linear flutter analysis has been explained in detail in previous works.<sup>4</sup> Classical linear flutter analyses are accomplished in the frequency domain and involve solving for the natural vibration frequencies and mode shapes, solving for the generalized aerodynamic forces, and finally solving for the modal damping and frequency variations with velocity.

Although several forms of differential equations of motion of a linear, forced vibration, viscously damped system are accepted<sup>16</sup>, the general practice of linear flutter analysis uses

$$m\ddot{q} + C\dot{q} + Kq = Q \quad (2-1)$$

where  $q$  is a displacement vector,  $m$  is the mass matrix,  $C$  is a damping coefficient matrix,  $K$  is a stiffness term matrix, and  $Q$  is an applied force. For free vibration analysis, no forcing function is included and damping is assumed to be zero. Equation 2-1 reduces to

$$(-m\omega^2 + \mathbf{K})\bar{\mathbf{q}} = 0 \quad (2-2)$$

and is solved using classical eigenvalue methods for  $\omega_i$ , the vibration frequency of the  $i^{\text{th}}$  mode, and  $\mathbf{q}_i$ , the mode shape (or modal amplitude) for the  $i^{\text{th}}$  mode. Solving the eigenvalue problem in this form yields the lowest frequency first, thus ensuring the greatest accuracy on the fundamental modes of interest.<sup>17</sup>

Since the process and derivation of the aerodynamic forces, namely the unsteady aerodynamic solutions, used in linear flutter analysis is not directly germane to the hybrid physical model proposed, it will be discussed in general as background. Refer to Denegri<sup>4</sup> and Cutchins' evaluation for more detail.

The flutter equations (Eq. 2-1) are solved using the Laguerre iteration method,<sup>18</sup> which is a variation of the classical k-method of flutter determinant solution. An unsteady aerodynamic solution is found by letting  $\mathbf{Q} = \mathbf{q}[\mathbf{AIC}]^*$ , computed using the doublet-lattice method.<sup>17</sup> In this method the wing surface is discretized into boxes. A constant pressure doublet filament of finite length is placed at the  $1/4$  chord of each box and the downwash is evaluated at the  $3/4$  chord mid-span of each box, thus the integration over the wing surface becomes a finite summation over the number of boxes. This process, including the definition and form of Kernel functions used to approximate the summation is a topic of research unto itself. The current convention is to approximate the Kernel by a complex parabolic function. The downwash is solved for providing  $[\mathbf{AIC}]$  for a specified Mach number and air density.

The free vibration modes, along with the aerodynamic influence coefficient matrix derived through the unsteady aerodynamic solution are then used to obtain a generalized

---

\* The three-letter matrix designation  $[\mathbf{AIC}]$  of the Aerodynamic Influence Coefficients follows the convention used in aeroelastic literature and is used here for continuity.

mass and generalized aerodynamic forces from which the final flutter solution is derived. The results are the flutter solution frequencies and dampings for each mode are plotted as functions of Knots Calibrated Airspeed (KCAS) (linear flutter analysis results for the flight used in this work will be shown in a later section of this chapter).

Since both the free vibration and unsteady aerodynamic solutions are solved in the frequency domain, and modal frequencies are of interest here, a more indepth evaluation of the original linear equation (Eq. 2-1) as it pertains to those frequencies will provide additional background.

Starting with the relevant basics, the differential equations of motion of a linear, forced vibration, undamped system are given by Eq. 2-1 with  $C = 0$ . Recall that it is standard practice to assume simple harmonic motion. The displacement vector is defined in terms of its magnitude and frequency as

$$\mathbf{q} = \bar{\mathbf{q}} e^{j\omega t} . \quad (2-3)$$

For the free vibration case (Eq. 2-2),

$$(-\omega^2 \mathbf{m} + \mathbf{k})\bar{\mathbf{q}} e^{j\omega t} = \mathbf{0}; \quad \therefore \quad (-\omega^2 \mathbf{m} + \mathbf{k})\bar{\mathbf{q}} = \mathbf{0} . \quad (2-4)$$

This can be cast as an eigenvalue/eigenvector problem by multiplying through by  $\mathbf{k}^{-1}$  and

$$\frac{1}{\omega^2}$$

$$(\mathbf{k}^{-1} \mathbf{m} - \frac{1}{\omega^2} \mathbf{I})\bar{\mathbf{q}} = \mathbf{0} , \quad (2-5)$$

Where the eigenvalues are

$$\lambda_i = \frac{1}{\omega_i^2} , \quad (2-6)$$

Providing the modal frequency for the  $i^{\text{th}}$  mode.

The eigenvectors  $\phi_i$  contain the modal displacement  $q_i$  for the  $i^{\text{th}}$  mode at each point in the DMM. This is know as the mode shape vector



$$\varphi_1 = \begin{bmatrix} \bar{q}_{1,1}^{fv} \\ \bar{q}_{2,1}^{fv} \\ \vdots \\ \bar{q}_{N,1}^{fv} \end{bmatrix}_{N \times 1} \quad (2-7)$$

The mode shape vectors are formed into a mode shape matrix  $\Phi$ ,

$$\Phi = \begin{bmatrix} \bar{q}_{1,1}^{fv} & \bar{q}_{1,2}^{fv} & \cdots & \bar{q}_{1,M}^{fv} \\ \bar{q}_{2,1}^{fv} & \bar{q}_{2,2}^{fv} & \cdots & \bar{q}_{2,M}^{fv} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{q}_{N,1}^{fv} & \bar{q}_{N,2}^{fv} & \cdots & \bar{q}_{N,M}^{fv} \end{bmatrix}_{N \times M} \quad (2-8)$$

for N wing locations and M modes.

Once the unsteady aerodynamic solution has been solved and the [AIC] matrix is available, it is possible to show the forced response of each point defined by the DMM. The transformation from modal space to the wing structural coordinates is made by

$$\bar{q}_{N \times 1} = \Phi_{N \times M} \mathbf{r}_{M \times 1} \quad (2-9)$$

The vector  $\mathbf{r}$  is defined as generalized coordinates in modal space and defines the modal deflection over the entire wing structure by scaling free vibration mode shapes into a final deflection vector resulting from structural characteristics and aerodynamic forces. Mathematically  $\mathbf{r}$  is the eigenvector resulting from the solution to Eq. 2-1 using generalized mass, generalized stiffness, and generalized aerodynamic forces.

This typically concludes the flutter analysis. But some additional information can be obtained with a bit more analysis. Using the flutter results as a basis, it is possible to obtain a time history of the actual generalized coordinates as they actually occur during a flight test. From that, the deflection measurements from a few accelerometers can be extrapolated to provide the deflection across the entire structure. Further, given the proposed hybrid physical model, it is possible to obtain a time history of each mode from actual flight.

Consider as a corollary to Eq. 2-9, the definition of a time-dependent deflection vector

$$\mathbf{q}(t)_{Nx1} = \boldsymbol{\varphi}(t)_{NxM} \mathbf{r}(t)_{Mx1}. \quad (2-10)$$

First, suppose that the actual deflections over the entire structure encountered during flight,  $\mathbf{q}'(t)_{Nx1}$ , are known. Further, each member  $q'(t)_n$  is made up of a linear combinations of modal deflections  $\phi'(t)_{n,m}$  as

$$\boldsymbol{\varphi}'(t) = \begin{bmatrix} \bar{q}'_{1,1} e^{j\omega_1 t} & \bar{q}'_{1,2} e^{j\omega_2 t} & \dots & \bar{q}'_{1,M} e^{j\omega_M t} \\ \bar{q}'_{2,1} e^{j\omega_1 t} & \bar{q}'_{2,2} e^{j\omega_2 t} & \dots & \bar{q}'_{2,M} e^{j\omega_M t} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{q}'_{N,1} e^{j\omega_1 t} & \bar{q}'_{N,2} e^{j\omega_2 t} & \dots & \bar{q}'_{N,M} e^{j\omega_M t} \end{bmatrix}_{NxM} \quad (2-11)$$

The vector  $\mathbf{q}'(t)$  can be defined by

$$\mathbf{q}'(t)_{Nx1} = \boldsymbol{\varphi}'(t)_{NxM} \mathbf{r}'(t)_{Mx1}. \quad (2-12)$$

The assumption must be made that the deflection at each point is a superposition of the modes, meaning that  $\mathbf{r}_{Mx1} = [1 \ 1 \ \dots \ 1]^T$ . This is a reasonable assumption, considering the purpose of  $\mathbf{r}$  is to apply the effects of the unsteady aerodynamic solution's forces to each mode's free vibration displacements. There is no need to do this for  $\mathbf{q}'(t)$ , since it is a measure of responses resulting from those forces.

Since the actual displacement is available, the next logical step would be to calculate an approximation for the generalized coordinate vector  $\mathbf{r}(t)$  that would cause the modeled wing displacements to match actual displacements due to the aerodynamic forces encountered during flight. In other words, it is desired  $\mathbf{r}(t)$  to find such that

$$\mathbf{q}'(t)_{Nx1} \cong \mathbf{q}(t)_{Nx1}. \quad (2-13)$$

The vector  $\mathbf{r}(t)$  can be found by

$$\mathbf{r}(t)_{Mx1} \cong \boldsymbol{\varphi}(t)_{MxN}^{-1} \boldsymbol{\varphi}'(t)_{NxM} \mathbf{r}'(t)_{Mx1}. \quad (2-14)$$

There is an exact solution to Eq. 2-14 when  $M = N$ . When  $M \neq N$  the Moore-Penrose<sup>19</sup> pseudoinverse can be used to approximate the inverse of  $\phi(t)$  if  $\phi(t)$  is at least of rank  $M$ .

The inverse of  $\phi(t)_{N \times M}$  is given by

$$\phi(t)_{M \times N}^{-1} = \left( \phi(t)_{M \times N}^T \phi(t)_{N \times M} \right)_{M \times M}^{-1} \phi(t)_{M \times N}^T. \quad (2-15)$$

When  $M < N$  the solution is overdetermined, and when  $M > N$  the solution is underdetermined. Although there are infinite solutions the Moore-Penrose algorithm seeks to minimize the norm of the solution vector, in this case  $\mathbf{r}(t)$ .

Since the actual surface deflection is usually unknown for all but a few of the  $N$  points, a method by which the wing deflection can be inferred from the available data is desired. Suppose a subset  $S$  containing  $P$  of the  $N$  surface points provides data for analysis. The deflection at those points is given by

$$\mathbf{q}'_S(t)_{P \times 1} = \begin{bmatrix} q'_1(t) \\ \vdots \\ q'_P(t) \end{bmatrix}_{P \times 1}. \quad (2-16)$$

For short periods of flight under constant conditions, it is desired that

$$\mathbf{q}'_S(t)_{P \times 1} \cong \mathbf{q}_S(t)_{P \times 1}. \quad (2-17)$$

If  $\mathbf{q}'_S(t)$  can be decomposed into a linear combination of its modal components as in Eq.

2-11, an approximation to  $\mathbf{r}(t)$  can be made from Eq. 2-12 as

$$\mathbf{r}(t)_{M \times 1} \cong \phi(t)_{M \times P}^{-1} \phi'_S(t)_{P \times M} \mathbf{r}'(t)_{M \times 1}, \quad (2-18)$$

with only the appropriate points in  $\phi(t)$  matching the  $P$  test point locations. The inverse of  $\phi(t)^{-1}$  is underdetermined, so although all modes are represented in  $\mathbf{r}(t)$  the modes with the highest energy dominate.

Equation 2-18 provides an approximation to the generalized coordinates for all  $M$  modes as a function of  $P$  data points. Using this an extrapolation of the actual deflection

of the wing surface can be made as a function of time by applying  $\mathbf{r}(t)$  found in Eq. 2-18 to the entire  $N \times M$  free vibration  $\phi$  matrix

$$\mathbf{q}_{extrap}(t)_{N \times 1} = \Phi(t)_{N \times M} \mathbf{r}(t)_{M \times 1}, \quad (2-19)$$

and can be used as a method by which the deflection of a surface as a function of time can be extrapolated from a limited number of measurement sights.

The DMM used in the linear flutter solution is a discrete-point damped spring-mass simplification of a finite element model of the aircraft wing (Figure 2-1). The flutter solution is solved for each point, and a vector of modal frequencies is determined. A matrix of maximum wing deflection values is provided for each modal frequency, at each DMM point on the wing.

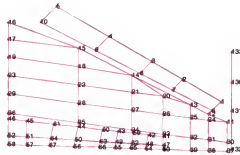


Figure 2-1 DMM points on a wing.

It is interesting to note the locations of the accelerometers with respect to the DMM points on the mass model wing. The forward wing tip accelerometer is closest to DMM point 133. In fact it can be assumed that DMM point 133 and the forward accelerometer are collocated for the purposes of this work. The aft wing tip accelerometer is located near but not exactly co-located with DMM point 132.

All analyses contained herein are performed using sea-level density and 0.95 Mach Doublet-Lattice method aerodynamic influence coefficients, but in general these

boundaries can vary from test to test. The free vibration analyses are performed for a half-airplane model using a matrix iteration method. The first 16 anti-symmetric flexible modes are retained for the flutter solution.

### **A Brief Flight Test Overview**

Historically, flight testing for the purpose of gathering data to measure the response of an aircraft system has been an expensive endeavor filled with uncertainties. On many occasions flights produced unexpected results, for instance an LCO or flutter response begins at flight conditions different than where they were expected to occur. On other occasions there have been portions of flight with no active flutter response when a one was expected, thus wasting the flight. Any deviation from the expected response during a flight translates into extra costs in both money and manpower.

In the U. S. Air Force, compatibility testing for a particular combination of an aircraft and store loading is accomplished by the Air Force SEEK EAGLE Office at Eglin Air Force Base, FL. Their process<sup>20</sup> for certifying a particular configuration for safety of flight includes engineering analysis and flight testing. Linear flutter analysis seeks to identify whether or not the configuration will result in a flutter or LCO response at a particular set of flight conditions. If their analysis indicates a possibility of such a response their practice is to fly a test aircraft properly loaded and instrumented under controlled conditions to either confirm the existence of the response or clear the conditions and configuration for flight.

Dreyer and Shoch<sup>21</sup> provides a complete overview of Eglin Air Force Base flutter testing. Testing is accomplished by flying the aircraft through a set of flight conditions that define an envelope, including conditions where active responses are likely to occur. Many times the aircraft is equipped with a flutter excitation system. This system is

designed to induce active vibration of the wing at a set frequency. By initiating a response at a benign condition engineers monitoring the flight can get an indication of how the aircraft might behave at more dynamic conditions and thus predict and avoid dangerous responses.

A typical flight can last several hours and cover a wide range of test conditions. It is not unusual for a single test flight to cover 50 to 75 individual test points, some covering nearly 100. These are usually functions of three independent variables: Mach number; normal acceleration, an acceleration in a direction normal to the flight path in units of  $1g \approx 32.2 \text{ ft/sec}^2$  or about the force of gravity at sea level, often referred to as g-loading or 'pulling g's'; and altitude. Roll is occasionally tested as well, though not as often. Normal acceleration is induced by transitioning from level flight to a climb, or by using a wind-up turn wherein the aircraft is banked and then turned at an increasing rate to generate an increasing g-loading. Test conditions are often grouped in rows on a test plan. These rows or lines usually contain points at a fixed Mach number and altitude and extend upwards through several values of normal acceleration. In-flight refueling is often necessary to extend the flight time of the aircraft.

Once a test condition has been reached the pilot attempts to maintain it for as long as necessary for engineers monitoring the flight to make an assessment of the response. Many times maintaining conditions is a difficult task since the condition may be highly dynamic and the response severe. Sets of abort criteria are established to ensure the safety of the crew and aircraft.

## **Flight Test Data Sets and Configuration Linear Flutter Analysis**

### **Aircraft Accelerometer Locations**

The aircraft is modified with accelerometers and data recorders located at strategic, but practical points on the aircraft. Since instrumentation is expensive and aircraft modifications difficult to make, the number and type of sensors is limited. There are several accelerometers on the test aircraft (Figure 2-2). For the purposes of flutter and LCO testing, the two on the wing tips are the most practical. There is a third, located on a store mounted under the wing but its location is less than desirable for measuring the response of the wing itself because it has secondary structures between the wing and the instrumentation that may modify or influence the measured response at that point.

The forward wing tip accelerometer is located approximately 4 inches aft of the wingtip launcher nose. The aft wing tip accelerometer is about 12 inches forward of the aft end of the launcher. The measurement range of the accelerometers is  $\pm 10g$  with a nominal sensitivity of 10 mV/g (at 100 Hz) and a noise floor of 0.0012g (rms). Data is acquired at 216.99 samples per second, using a programmable data acquisition system. The Nyquist frequency is about 108 Hz at that sample rate. This is well above the frequencies of interest, as well as their lower harmonics. Linear analysis typically predicts the first few modes to occur below 30 Hz, with most of the oscillation energy below 20 Hz, so the frequency band is sufficient. The aircraft instrumentation system contains an internal low pass filter whose cutoff frequency is purported to be around 120 Hz. Unfortunately the characteristics of this filter are not available. It is therefore assumed that any phase shift caused by the filter is consistent throughout the flight and the data is taken as truth.

The flutter excitation system installed on the test aircraft is designed to excite vibrational modes in the aircraft structure by introducing an input signal to the flaperon servoactuators, which would move the flaperon a maximum of  $\pm 1$  degree at frequencies from 2 to 20 Hz. This system makes it possible to excite and identify specific vibration modes at successively higher speeds to accurately determine damping or structural stability at a specific test condition<sup>1</sup>.



Figure 2-2 Aircraft configuration and sensor locations.

### **Mission Particulars**

Data for this work is from mission 8951, flown on May 28, 1999 at Eglin Air Force Base, Florida. The aircraft was symmetrically loaded with empty wing tip launchers, missiles on the under-wing station, a 500lb weapon on the outer weapon stations, and a fuel tank on the inner weapon stations (Figure 2-2). The flight was designed to measure the response of the aircraft under flight conditions ranging from 0.65 Mach to 1.05 Mach, at 2000 feet, 5000 feet, and 10,000 feet, and at g-loadings of 1g, 2g, ... 6g. The test plan



showed about 60 individual test points to be evaluated, organized into 20 lines. Flutter or LCO was expected during the flight and an abort criteria was set as sustained or increasing vibration at 7 Hz with an amplitude of 3.0g or higher.

### **Linear Analysis of Flight Test Configuration**

Results of linear flutter analysis of the aircraft configuration used in Flight 8951 is shown in Figures 2-3 and 2-4. Predicted damping (Figure 2-3) is shown for each mode as a function of KCAS. The term 'knot' refers to speeds with units in nautical miles per hour. Figure 2-4 shows the predicted frequencies associated with each mode, also as a function of KCAS. There are subjectively five 'active' modes as determined by reviewing the linear flutter analysis outputs; Modes 1, 2, 8, 11, and 15, as well as several benign modes, with Mode 10 appearing to be the least active mode. Active modes are those that indicate a definite damping trend, either positive or negative.

Mode 2 is the most active mode and is the only mode predicted to have positive damping. Notice that its damping increases dramatically at around 350 KCAS. Its frequency at that speed, read as an approximation from the chart in Figure 2-4 is 7 Hz. The corresponding free vibration analysis frequency for that mode is 7.01 Hz. It was therefore expected that a 7 Hz flutter response would be experienced on this flight beginning at an airspeed approximate to 350 KCAS. The relationship between KCAS,<sup>22</sup> shown as curved lines, Mach number and altitude is shown in Figure 2-5. Since the aircraft's operational range includes mid and transonic airspeeds, vibrations that are expected at speeds at or greater than 350 KCAS influence a large portion of flight as can be seen in the plot.

The free vibration analysis frequencies for all modes are given in the Table 2-1 for this flight.

Table 2-1 Flight 8951 free vibration analysis frequencies by mode.

Mode Number	1	2	3	4	5	6	7	8
Frequency (Hz)	6.55	7.01	8.42	8.58	9.80	12.00	14.20	14.70
Mode Number	9	10	11	12	13	14	15	16
Frequency (Hz)	15.20	16.50	17.90	19.40	23.00	26.20	27.00	30.20

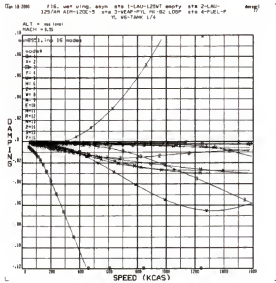


Figure 2-3 Linear flutter analysis of test configuration, Damping.

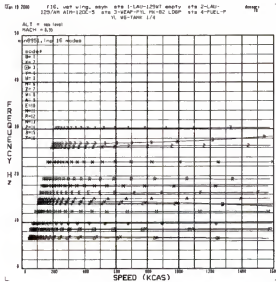


Figure 2-4 Linear flutter analysis of test configuration, Frequency.

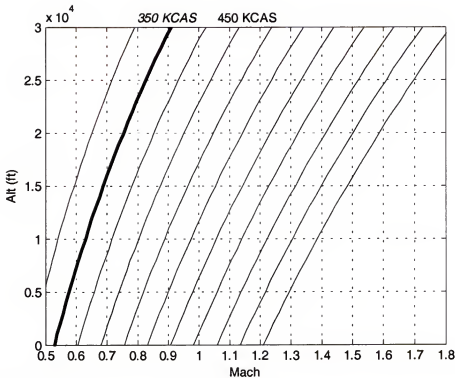


Figure 2-5 KCAS, Mach, and altitude relationship.

### Flight Test Signals

Test signals for this work were taken from two portions of the flight. Data from both the forward and aft accelerometers from the left wing were used. The first, Test line 17, is a signal showing increasing LCO with linearly increasing Mach number. The second, Test Points 8, 9, 10, 11, 12, combined as Test line 3, are data sets showing nonlinear LCO behavior with increasing g-loading during a wind-up turn.

#### Test Line 17 Test Signal

Test line 17 is a portion of flight about 25 seconds long at 5000 feet Mean Sea Level (MSL), in level flight (1g), with the Mach number ramping up from 0.78M to 0.9M over about 20 seconds. Limit cycle oscillation was observed during this test with increasing amplitude in proportion to increasing Mach number. The flight conditions (test states) are shown in Figure 2-6. Altitude in feet, normal acceleration in g's and Mach

number are shown as functions of elapsed time for that portion of flight. The test plan was for level flight. It is clear from the plot that the pilot was able to maintain near-constant altitude for the entire test duration within a 75 foot altitude band.

The normal acceleration plot shows that there was some deviation from the planned value. This is due to normal flight turbulence and is an unfortunate reality of flight testing, resulting in higher noise levels and variations in the data.

Mach number was controlled very well, increasing at a nearly linear rate from about 0.78M to 0.9M over 20 seconds of flight. The Mach number remains constant after 20 seconds.

Figure 2-7 shows the forward accelerometer signal for this portion of flight, along with its Power Spectral Density (PSD). As predicted, the response is a form of flutter. In the beginning of the segment we see that LCO is already present at low but sustained levels. The amplitude of the oscillation increases steadily with increasing Mach number and appears to be flutter at first glance. But a close inspection of the figure shows that as Mach number becomes stable around 0.9, so too do the oscillation amplitudes. The last five seconds or so are steady, beating limit cycle oscillations with large maximum amplitudes—around 3.5 g's. This was at the abort criteria, but since the response was LCO rather than true flutter the test point was maintained as long as it was deemed safe to do so. The test point was finally terminated five seconds after reaching an airspeed of 0.9M.

The primary mode of vibration has a frequency of about 7 Hz as predicted, shown on the PSD in figure 2-7. A second rise in energy is noted at about 14 Hz for this signal. In fact this 14 Hz artifact is noticed in all responses throughout the flight on the forward sensor. Linear flutter analysis predicted a mode at 14 Hz, but did not predict it as an

active mode. In fact the three modes in the neighborhood of 14 Hz, Modes 6, 7, and 8, all were predicted to have negative damping. Consequently it was a surprise to engineers monitoring the flight to see this mode appear.

The signal from the aft sensor for the same portion of flight is shown in Figure 2-8, along with the corresponding PSD. Notice the absence of the 14 Hz component. Also, the signal tracing does not exhibit the beating tendencies of the signal from the forward sensor. Intuitively there is some explanation for the difference between the two signals since the forward sensor is attached towards the end of what can be considered a long, thin protrusion that would be more susceptible to harmonics. Conversely the aft sensor is mounted on a portion of the launcher that is attached directly to the wing. The structural integrity at the sensor location is much stronger, so it seems logical that vibration amplitudes would be lower and higher frequencies more likely to be damped. This assertion is consistent with the signal tracings and PSD in Figure 2-8.

The nearly steady flight conditions and the way the LCO response seemed to be in correlation with increasing airspeed made this a good candidate for data to challenge the proposed architecture and sequential neural network process. The results of experiments will give insight the modal content and nature of the wing' s response as a function of Mach number as the one independent variable.

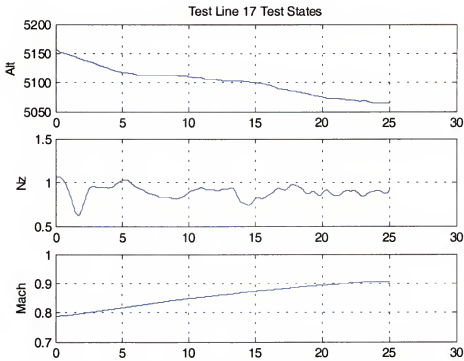


Figure 2-6 Test line 17 sensor traces: altitude, Mach and normal acceleration.

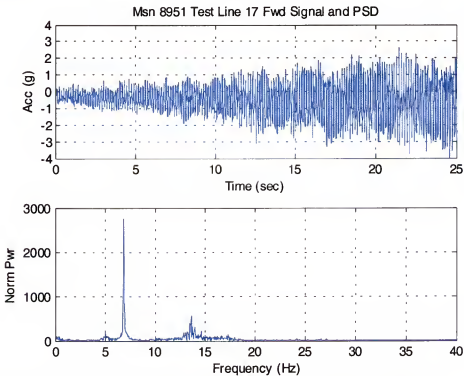


Figure 2-7 Test line 17 forward wing tip signal and power spectral density.

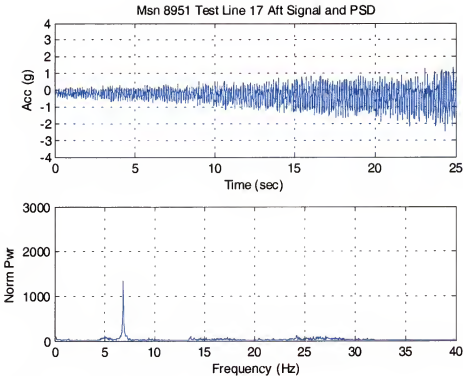


Figure 2-8 Test line 17 aft wing tip signal and power spectral density.

### Test Line 3 Test Signal

In order to assess the robustness of the proposed system for data synthesis a second set of flight conditions was chosen from the same mission. Earlier in the test a set of conditions was flown over a range of g-loadings. Test line 3 was planned for an altitude of 10,000 feet, at 0.8M, with normal acceleration test points from one to five g's in one-g increments, organized into five tests, Test 8, 9, 10, 11 and 12. A wind-up turn was used to fly the aircraft in a circle with a decreasing radius in order to increase the loading on the aircraft. Figure 2-9 shows the flight conditions as well as the forward (labeled Fv1) and aft (Av1) accelerometer outputs as functions of elapsed time from the beginning of the test in seconds.

The first portion of Test line 3, Test 8, is unique in that it contains three distinct portions of flight. Not long into the test, the flutter excitation system was activated for approximately 2 seconds. A mild LCO response was present at the beginning of the test. The excitation system was used to elevate the response in order to observe whether or not damping of the signal would occur. The LCO response rose to nearly 3g's during the excitation and damped back to the previous level after the excitation was terminated. The portion of the response after excitation is labeled Test 8c for this work and will be used in experiments designed as an initial evaluation of the LLM paradigm. The Test 8c left wing, forward missile tip accelerometer response is shown in Figure 2-10.

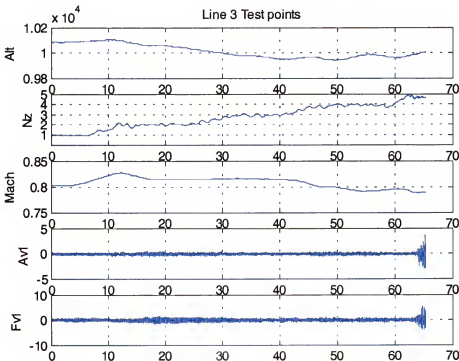


Figure 2-9 Test line 3 flight test states and signals.

The pilot was able to maintain altitude within a band of about 200 feet during Test line 3. Normal acceleration was increase over the 65 second interval from 1 to about



4.5g's, but not as consistently linear as was the increase in Mach number in the Test line 17 data. The goal was not to measure response over a steadily increasing load, but instead to fly the aircraft to a condition and hold it briefly for evaluation. The aircraft was then flown to the next condition. This can be seen in the Nz plot. There are short periods of time where the g-loading is constant, followed by changes to the next loading value. Also note that Mach number is not as consistent. It is evident, though, that once the desired test condition was attained the parameters were held constant for as long as possible. This becomes increasingly difficult as g-loading increases. Finally the LCO responses were becoming dangerously large as the g-loading reached 4.5g's and the test was terminated.

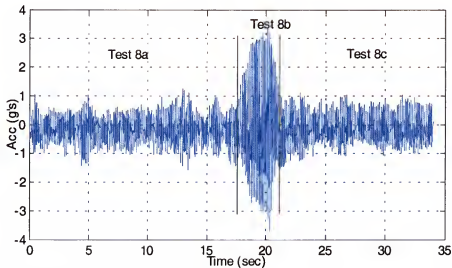


Figure 2-10 Test 8 forward wing-tip accelerometer.

Since the flight test states between test conditions are somewhat chaotic and are only consistent during shorter portions of flight at the planned test conditions, the data from Test line 3 was divided into short segments, each to be evaluated separately. The intent is to create models for each as a function of normal acceleration. It is assumed that the other flight parameters remain constant during that portion of flight. The test

segments are illustrated in Figure 2-11. Signals from both forward and aft accelerometers are taken at portions of flight where Mach, altitude and normal acceleration are nearly constant and at the intended test point condition. The test data set includes the flutter response measured at 4.5g's.

The response of the wing during these test points does not display the linear tendencies of Test line 17's data. There appears to be some increase during the sequence, but it is difficult to assign one particular flight test parameter to its cause. The exception to this is the flutter response at the end, where it is noted that as g-loading increased beyond 4g's the response became unstable. The PSD for these signals are shown in Figure 2-12. The 7 Hz component is still the primary mode, and the forward accelerometer is once again displaying the 14 Hz subcomponent. As with the aft sensor data from Test line 17, the 14 Hz vibrations are very weak. This is an indication that the modal content of the wing response is the same throughout the flight, at least in their frequencies. These signal characteristics make the data from Test line 3 attractive as a further test case for the proposed data synthesis system since it is can model each separately.

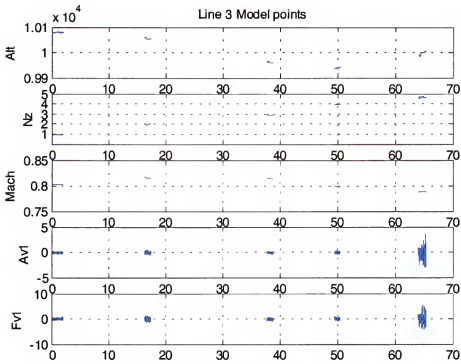


Figure 2-11 Test line 3 analysis points, flight test states and signals.

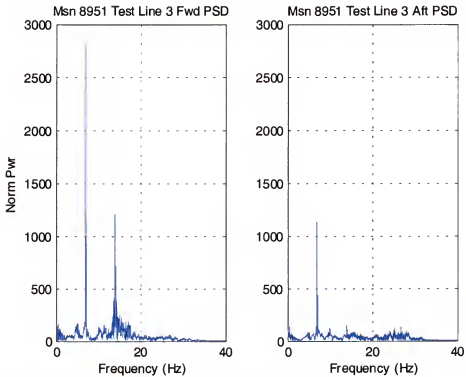


Figure 2-12 Test line 3 PSD, forward and aft sensors.

## Alternative Approaches

### Engineering Approaches

Research involving aeroelastic characteristics of thin wing aircraft is far from bridging the gap between theory and reality. There are many methods employed by researchers to further the understanding of aircraft limit cycle oscillation and flutter. Many are actively seeking better ways of modeling wings, and simulating the response and characteristics of wings and their different configurations. Linear analysis has been at the forefront, although it is known that the LCO is a nonlinear phenomenon. Several researchers are working to better define the nonlinearities that may play a part in the responses of wings.

It is understood that flutter and LCO are phenomenon caused by a combination of structural dynamics and aerodynamics, and that the specific combination of these forces changes with each wing and wing/weapon configuration.<sup>23</sup> Linear flutter analysis has been discussed in detail. What follows is a brief survey of other current analysis methods and related research.

### Volterra series

There are many efforts being made to apply nonlinear methods to predicting aeroelastic responses of wings. Silva<sup>24,25</sup> et al. at NASA Langley Research Center, and Raveh<sup>26,27</sup> et al. at Georgia Institute of Technology are focusing their aeroelastic research on Reduced-Order Models (ROM) based on a Volterra series. Silva is using ROM's to model nonlinear behavior in a wing surface to speed up CFD simulations. Raveh is using them directly to model the responses of surfaces. Both draw on data obtained from CFD analysis, and use either first or second order Volterra series to identify in impulse or step response in a surface. A second order Volterra series, shown in Eq. 2-20 is very similar to the convolution of two frequency responses. Contrasted to this is the LLM system

proposed in this work, which effectively changes the frequency response locally as necessary, but provides global synthesis. The Volterra series second-order term looks at a single system response at two different times, making it a broader but still global model in time method. These works demonstrate successful approaches in LCO prediction by using nonlinear system response techniques. Their limitation is that they are fixed-coefficient global models based on computer generated data. As such they aren't yet able to bridge the gap to real world aircraft data.

$$y(n) = h_0 + \sum_{k=0}^N h_1(n-k)u(k) + \sum_{k_1=0}^N \sum_{k_2=0}^N h_2(n-k_1, n-k_2)u(k_1)u(k_2). \quad (2-20)$$

### Nonlinear empirical methods

A very sizable amount of research is being conducted into the physical cause of wing flutter and LCO, and subsequent empirical methods for modeling the phenomenon in closed form solution. Cunningham<sup>28</sup> casts conventional LaGrangian-form equations of motion in terms of structural vibration modes, with mass, stiffness and structural damping terms included. All aerodynamic forces take the form of generalized forces that are time dependent and nonlinear. The static aeroelastic solution needed for use with the dynamic equations of motion is an approximation that is obtained by iteratively solving the nonlinear reduced equations of motion with all time-varying terms removed. The nonlinear aerodynamic forces are calculated at rigid static mean conditions as the first iteration and in the following steps at statically deforming mean conditions until the solution converges.

Cooper and Wright<sup>29</sup> use Normal Form Theory to predict the amplitude of LCO as functions of airspeed and the reduced frequency. Their approach is analytical and does away with the need for numerical simulation but is limited to continuous nonlinearities, and two degree-of-freedom systems. The nonlinearities are modeled as a second order

dependence on velocity in the response, and cubic stiffness in the forcing function. An in-depth evaluation of these and other efforts is beyond the scope of this work, but it's important to include the topic in the general discussion of flutter and LCO prediction, since it lies at the very heart of the problem, and is the ultimate goal of most flutter and LCO research.

### **NASA solutions**

One important example of physical testing is the work being done at NASA Dryden Flight Research Center.<sup>30</sup> Lind has developed a small, instrumented wing that mounts on the centerline of an F-15. The advantage of the F-15 is that it has much more stable flutter properties. It can be flown at much higher speeds, allowing flutter and LCO testing to be accomplished at specific test points that may be beyond the limits of other thin-winged aircraft.

NASA has begun characterization of the wing using ground vibration techniques. They have expanded this technique to include additions to existing uncertainty models.<sup>31</sup>

### **Similar ANN work**

There are several ongoing efforts to predict the various phenomenon related to flutter and LCO using artificial neural networks. Faller<sup>32,33</sup> is following two tracks: one to look at errors in measurement instrumentation; the other using ANNs to predict wing point pressures. Cooper<sup>34</sup> is looking at predicting damping coefficients to existing equations using ANNs with pseudo-data. Wong's effort<sup>35</sup> is very similar to the proposed method, except he has simplified his model, and is using artificially derived signals, rather than actual measured data. Cunningham<sup>28</sup> is using classical methods, but is using simple summed-modes as inputs to his prediction in a manner very similar to that in this work.

Related, but perhaps further separated is the work done by McMillen,<sup>36</sup> who used ANNs to predict unmeasured flight test states for a given flight condition, such as angle of attack. This was a feasibility study that used modeled flight data as its training signal, but showed that it was possible to predict flight states with a standard, feedforward ANN. Another feasibility study showed that ANNs could be used to predict the flutter speed of a modeled general aviation aircraft. Although the data used was simplified model outputs, the flutter speed was shown to be predictable, once again using a standard, feedforward ANN.<sup>37</sup>

There has also been some work in the use of intelligent systems for controlling flutter during flight, although not as applicable to this proposal, since the aircraft used for USAF flutter and LCO testing are no longer in the design stage. Therefore active control systems cannot be added. Lichtenwalner's work, however, shows that given the accessibility of control systems, active flutter suppression can be possible using ANN's.<sup>38</sup>

## **Physical Modeling Approaches**

### **Computational fluid dynamics**

Computational fluid dynamics has been an essential part of flutter<sup>39</sup> and LCO analysis since its inception, especially in the past few years, as computer speed has increased sufficiently to begin making the technology more practical. Unfortunately, CFD solutions are still very computationally intensive. In order to reduce the computational time, the aircraft and aerodynamic models and properties used must be simplified to the point where the resultant flutter solution is still less than desirable. A complete discussion of CFD as it applies to flutter is certainly beyond the scope of this dissertation. But it is interesting to review some of the very basic concepts in order to see

the advantages and limitations of the technology in comparison to the proposed hybrid physical model.

An obvious difficulty with a nonlinear process is that the concept of eigenvectors and eigenvalues, while valid for linear aeroelastic modeling, may no longer apply in a nonlinear aeroelastic system. Computational Fluid Dynamic modeling is a method by which nonlinear aerodynamic responses are computed. Because these models solve nonlinear equations using time-marching analysis, the nonlinear flutter analyses are performed in the time domain.<sup>39</sup>

In the time-marching analysis the aerodynamic forces are coupled with the equations of motion through a vector of generalized aerodynamic forces, and the coupled aeroelastic system is marched in time, performed in two phases. First, the flow field is converged to a steady state. Then initial boundary conditions (modal displacements or velocities) are prescribed to each mode separately and the time integration begins. There are several methods for determining these modes, one of which is a free vibration analysis as is done in linear flutter analysis. The transients for each mode determine the stability of the system. In order to determine the flutter conditions for a given free stream Mach number, aeroelastic transients must be computed at several values of dynamic pressure. The repeated numerical simulations required make the time-marching analysis method an extremely expensive process.

The approach is to transmit the nonlinear aerodynamic responses from the CFD model into the linear structural system (interesting to note is that often this is the same linear structure from which the data for linear flutter analysis is derived). The structure then responds to these forces with a deformation or motion, which is then passed back to the CFD mode for computation of a new aerodynamic load. This closed-loop iteration



continues, providing a time history for analysis. These analyses are performed at each Mach number, where the dynamic pressure is varied until an unstable transient is encountered, identifying the flutter condition. Obviously the repetitive execution of the CFD codes is costly and time consuming.

Linear stability analysis is a second method by which the nonlinear equations are solved, where the flutter conditions are determined from a traditional eigenvalue analysis using nonlinear Generalized Aerodynamic Force (GAF) coefficient matrices that are evaluated at different values of reduce frequencies. Analysis using GAF is nonlinear in the sense that they are used to compute the response to small-amplitude modal excitation about the steady-state nonlinear flow condition, retaining the nonlinear properties of the mean flow, unlike their linear counterparts that are independent of the mean flow.

The forced-harmonic method is a direct method for evaluating the nonlinear GAFs. Each mode is separately excited in a forced harmonic oscillation. Time-dependent boundary conditions are prescribed in the CFD analysis and the flow variables are iterated through a few cycles of forced motion. For flutter analysis, each mode has to be excited over a range of reduced frequencies. This is also, therefore, a very expensive method. There are many other variations on the implementation and evaluation of GAFs that are beyond the scope of this work. It is sufficient to understand that CFD, like the proposed method, relies on predetermined structural information as its starting point, and that this information is the same in many instances. The advantage of CFD is a highly accurate solution for a given flight condition. That accuracy comes at such a high cost, though that it is still impractical as a tool for production aircraft compatibility analysis.

Research in aircraft flutter prediction using CFD is being accomplished at NASA Langley Research Center,<sup>24</sup> as well as the Air Force SEEK EAGLE Office, Eglin AFB,

Fl.<sup>19</sup> Work by several other researchers attempting to reduce the computational complexity of the method, and identify the nonlinearities involved in flutter and LCO have taken advantage of CFD as an analysis tool<sup>40</sup> as well. Chen<sup>41</sup> has produced both flutter and LCO signals in a two-dimensional airfoil using CFD with ROMs, and is making progress towards improving more complex models.

### **Local linear models**

Local linear models will be discussed in the next chapter. By way of introducing the topic as a dynamic modeling approach, the two citations from Chapter 1 are briefly illustrated here. The work done by Principe<sup>10</sup> et al. uses LLMs mapped by Self Organizing Maps (SOM) to model wind tunnel dynamics as the position of the object in the tunnel is changed. They were able to show very good fine-resolution Mach number control by training a set of models and switching to the appropriate model according to the measured Mach number history. Each SOM models the forced tunnel dynamics and can be thought of as an autonomous wind tunnel. This is a very good example of the usefulness of LLMs for modeling a very active dynamical system.

A second applicable work is by Kohlmorgen<sup>42</sup> et al. wherein the concept of drifting dynamics and the use of log-likelihood algorithms to detect changes in a response sufficient to require a change in local model is described. He presents a method for unsupervised segmentation and identification of nonstationary drifting dynamics. He demonstrated that his method was applicable both drifting and switching time series.

Kohlmorgen uses ANNs with Hidden Markov Models for his work, but the concept is applicable here. The LCO response observed during many portions of flight can be described as a drifting rather than a switching response since the system is responding to a smooth change in state, or as a response transitions from a forced response to a damped

or steady state response. The results of his works are applicable since they demonstrate that data segmentation and LLMs can be used to describe a time series as a function of dynamics in general, not necessarily across a given state space but instead across temporal space. That is the approach taken in this work and will be addressed in the next chapter.

### CHAPTER 3 MODELING APPROACH

The proposed method for flight test data modeling requires several analyses for its development. This chapter begins by developing the LLM from portions of the linear flutter analysis governing equations. Evaluation of these equations will show that the model can be made of two consecutive feedforward parts, breaking the original ARMA system into separate AR and adaptive MA components. As will be shown, the AR physical model can be described as a SIMO system; a parallel array of oscillators initialized to the flutter frequencies of the structure. The MA component will consist of adaptive FIR filters arranged as a MISO system receiving inputs from the oscillators. The design of a practical oscillatory neural network architecture is required and will be provided in the second portion of this chapter. Having then formed the basic components of a model the transfer function of the system is developed. Third, the topic of LLMs in general will be briefly discussed. The chapter concludes by evaluating a method for unsupervised desired response segmentation, based on a modified version of the GLRT.

As introduced in Chapter 1, the basic underlying assumptions required for the development of the model are: that the response can be approximated by simple harmonic motion; that the response can be considered stationary and time invariant for the period of time required for adaptation; and that the modes do not converge.

#### **LLM Approach: Quasi-Fixed IIR/Adaptive FIR Modular Approximation of an IIR System**

As introduced in Chapter 2, the accepted governing equation for practical linear flutter analysis is

$$m \frac{d^2 \mathbf{q}}{dt^2} + \mathbf{C} \frac{d\mathbf{q}}{dt} + \mathbf{K} \mathbf{q} = \mathbf{Q}, \quad (3-1)$$

where:

$\mathbf{m}$  = mass, a diagonal matrix containing the mass for each DMM point,

$\mathbf{C}$  = damping coefficients,

$\mathbf{K}$  = stiffness, a diagonal matrix containing the stiffness coefficient for each DMM point,

$\mathbf{q}$  = wing vertical deflection, an  $N \times 1$  vector containing deflection at each DMM point and

$\mathbf{Q}$  = aerodynamic forcing function in vector form, the force at each DMM point, an  $N \times 1$  vector.

The assumption of simple harmonic motion in only the vertical direction reduces the description of each DMM point into independent spring-mass systems. The result is a simplification of mass and stiffness matrices so that they now have values corresponding only to the points themselves with no interaction and the matrices become diagonal.<sup>4</sup>

The transfer function of the IIR system in Eq. 3-1 can be determined by its Laplace transform. Consider first the case where there is only one point on the wing, and therefore only one mode. Recall that damping is considered zero and that  $\mathbf{Q}$  can be represented by the product of an Aerodynamic Interference Coefficient [AIC] and the displacement  $q$

$$m \frac{d^2 q}{dt^2} + Kq = AICq. \quad (3-2)$$

The transfer function for this equation is

$$\frac{AIC}{ms^2 + K} = \frac{AIC}{\sqrt{mK}} \left( \frac{\sqrt{\frac{K}{m}}}{s^2 + \frac{K}{m}} \right). \quad (3-3)$$

The inverse Laplace transform is

$$\frac{AIC}{\sqrt{mK}} \sin\left(\sqrt{\frac{K}{m}}t\right). \quad (3-4)$$

This shows that the behavior of a single point on the wing can be modeled as a scaled sine wave as expected.

Laplace analysis of the multiple mode case of the linear flutter solution is a bit more complex, but the results are the same. Beginning with Eq. 3-1 again, with  $C$  considered 0 and  $m$  and  $K$  now diagonal matrices, and replacing  $Q$  with  $[AIC]q$ , where  $q$  is now a matrix of displacements of each point with respect to all the other points on the wing given by

$$q = \begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,16} \\ \vdots & \ddots & & \\ \vdots & & \ddots & \\ q_{16,1} & & & q_{16,16} \end{bmatrix}, \quad (3-5)$$

(16 modes and 16 points are shown in Eq. 3-5, but this is not a limitation), Eq. 3-1 and its Laplace transform becomes

$$m \frac{d^2 q}{dt^2} + Kq = [AIC]q \xrightarrow{L} s^2 m Q(s) + KQ(s) = [AIC]Q(s). \quad (3-6)$$

The transfer function can be written as

$$H(s) = [s^2 m + K]^{-1} [AIC]. \quad (3-7)$$

The matrix  $[s^2 m + K]^{-1}$  is

$$[s^2 m + K]^{-1} = \begin{bmatrix} s^2 m_1 + K_1 & 0 & \cdots & 0 \\ 0 & s^2 m_2 + K_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & s^2 m_{16} + K_{16} \end{bmatrix}^{-1}. \quad (3-8)$$

As such the inverse is easily calculated. The product  $[s^2\mathbf{m} + \mathbf{K}]^{-1}[\mathbf{AIC}]$  can be shown to be

$$[s^2\mathbf{m} + \mathbf{K}]^{-1}[\mathbf{AIC}] = \begin{bmatrix} \frac{AIC_{1,1}}{s^2m_1 + K_1} & \frac{AIC_{1,2}}{s^2m_1 + K_1} & \dots & \frac{AIC_{1,16}}{s^2m_1 + K_1} \\ \frac{AIC_{2,1}}{s^2m_2 + K_2} & \frac{AIC_{2,2}}{s^2m_2 + K_2} & \dots & \frac{AIC_{2,16}}{s^2m_2 + K_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{AIC_{16,1}}{s^2m_{16} + K_{16}} & \frac{AIC_{16,2}}{s^2m_{16} + K_{16}} & \dots & \frac{AIC_{16,16}}{s^2m_{16} + K_{16}} \end{bmatrix}, \quad (3-9)$$

for a 16 point wing analysis. Realizing that  $\frac{A_m}{s^2m_m + K_m} \xleftrightarrow{L^{-1}} \frac{A_m}{\sqrt{m_m K_m}} \sin(\sqrt{\frac{K_m}{m_m}}t)$ , the

16-by-16 system response for Eq. 3-1 is

$$h(t) = \begin{bmatrix} \frac{AIC_{1,1}}{\sqrt{m_1 K_1}} \sin(\sqrt{\frac{K_1}{m_1}}t) & \frac{AIC_{1,2}}{\sqrt{m_1 K_1}} \sin(\sqrt{\frac{K_1}{m_1}}t) & \dots & \frac{AIC_{1,16}}{\sqrt{m_1 K_1}} \sin(\sqrt{\frac{K_1}{m_1}}t) \\ \frac{AIC_{2,1}}{\sqrt{m_2 K_2}} \sin(\sqrt{\frac{K_2}{m_2}}t) & \frac{AIC_{2,2}}{\sqrt{m_2 K_2}} \sin(\sqrt{\frac{K_2}{m_2}}t) & \dots & \frac{AIC_{2,16}}{\sqrt{m_2 K_2}} \sin(\sqrt{\frac{K_2}{m_2}}t) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{AIC_{16,1}}{\sqrt{m_{16} K_{16}}} \sin(\sqrt{\frac{K_{16}}{m_{16}}}t) & \frac{AIC_{16,2}}{\sqrt{m_{16} K_{16}}} \sin(\sqrt{\frac{K_{16}}{m_{16}}}t) & \dots & \frac{AIC_{16,16}}{\sqrt{m_{16} K_{16}}} \sin(\sqrt{\frac{K_{16}}{m_{16}}}t) \end{bmatrix}. \quad (3-10)$$

The motion at each wing point  $p$  is a sum of its corresponding column

$$q_p(t) = \sum_{n=1}^{16} h_{n,p}(t). \quad (3-11)$$

This can be expressed as

$$q_p(t) = \sum_{m=1}^{16} \frac{AIC_{p,m}}{\sqrt{m_m K_m}} \sin(\sqrt{\frac{K_m}{m_m}}t). \quad (3-12)$$

Equation 3-12 shows that the deflection of a DMM point on the wing is a linear combination of the modes. As such the modes can be treated in parallel by a system that first generates the modal signals, properly placing the poles of the system, then adjust each mode's amplitude as necessary. Sinusoidal signals can be provided by signal

generators, initialized to free vibration frequencies for optimal structural representation. Allowing the frequency of each mode to be adjustable through adaptation will allow for small variations in frequency, accounting for small pole movements. Modal amplitude scaling is then done by adaptable gain filters. The model output is the sum of the filtered sinusoids. Further, by increasing the filter order, phase as well as gain for each mode can be adjusted. The model is feedforward and can easily be divided into two modules; one to act as signal generators and the other a bank of FIR filters. A conceptual diagram of the model is depicted in Figure 3-1.

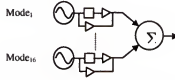


Figure 3-1 Conceptual diagram of oscillator driven FIR filter system.

A time domain representation of the output of this system is

$$y(t) = \sum_{m=1}^{16} a_m \sin(\omega_m t + \phi_m), \quad (3-13)$$

where  $m$  is the mode number taken from the first 16 solutions of the linear flutter analysis and  $\omega_m$  and  $\phi_m$  are the frequencies and phases for each mode. Recall that linear flutter analysis provides modal frequencies and amplitudes for all points analyzed on a wing surface, and selects the first 16 modes found in order of their frequency values to be the most active modes. Also, recall that relative phases of these modes are not identified. This is no longer a limitation with the new model since relative phase between modes can be established with FIR filters with at least one delay.



### Model Implementation

The design of the FIR filters is very straightforward. Single-delay filters are shown in Figure 3-1 since they are sufficient for adjusting both phase and gain of each mode, but the filter order is can be greater, as can the value of each delay. For this work a typical adaptable FIR filter will be implemented for each mode as

$$y(n) = W_0x(n) + W_1x(n-1) + \dots + W_Nx(n-N). \quad (3-14)$$

The weights  $W_0, W_1, \dots, W_N$ , where  $N$  is the filter order, are adaptable.

Since the LLM calls for an array of signal generators, the design of a simple oscillator is necessary. The design begins with a look at existing methods.

#### Kuo's Two-PE Oscillator as a Foundation

Several authors have approached the challenge of designing a small, stable, and adaptable oscillatory network for different applications.<sup>43,44</sup> Jyh-Ming Kuo<sup>45</sup> proposed a simple two-Processing Element network that can be trained to behave as an oscillator. He used Real Time Recurrent Learning (RTRL) to train the network to follow simple harmonic motion. Since these oscillators are recursive by nature, they must be trained using recursive methods, either Back Propagation Through Time (BPTT) or RTRL. Principe<sup>46</sup> et al. discuss both of these methods thoroughly. Since the front-end of the LLM in this work is recursive, the entire system must be trained using a recursive method. Kuo's results illustrate a challenge to this work in that the solution space for these oscillators has many local minima, and training may prove to be difficult.

Kuo's oscillator is used as a basic model for the development of a simpler two-PE linear oscillator. The phenomenon of oscillation in neural networks has been the study of researchers for many years. It is known that simple, two-Processing Element (PE) networks can be arranged in such a way that the output approaches sinusoidal oscillation

over a band of inputs and outputs, with varying configurations of weights, biases, node gains and delays.<sup>44,45,47</sup> A generic fully connected two-PE network is shown in Figure 3-2. At least one of the paths must be an inhibitor in order for the output to have oscillatory properties, and in most cases the PE's employ sigmoidal functions to ensure stability. Training is accomplished with variations of time-dependent back propagation algorithms.

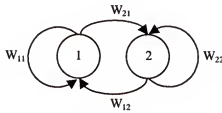


Figure 3-2 Kuo two-PE oscillator network.

For the purposes of this work, modifications to the existing methods are required. It is desirable that the oscillation frequency be controlled by adapting only one weight, since it allows the final adapted frequency to be calculated directly. Plasticity in a single variable also ensures that the output will remain sinusoidal until saturation is reached in the weight, a feature not found in higher degree-of-freedom oscillatory circuits. Speed and ease in training are also positive aspects of a single weight network. It is also desirable that the system be independent of a particular input sequence. A robust design would be a system that is infinite impulse response by nature, that ideally would respond indefinitely to a single impulse at the beginning of the sequence, and be self sustaining from then on, thus freeing memory and time keeping track of and sustaining complex input sequences.

If linear PEs are used in the generic recursive system shown above, it can be modeled in a state variable sense, as shown in Eq. 3-14. Differential equations describing

the behavior of a system are formed into functions of the states,  $\mathbf{x}$ , with the output described by  $\mathbf{y}$ .

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}\mathbf{u} \\ \mathbf{y} &= \mathbf{c}\mathbf{x}\end{aligned}\tag{3-15}$$

The coefficients for the state equations can be written as shown in Eq. 3-16. The matrix  $\mathbf{A}$  contains the four network weights, and the vectors  $\mathbf{b}$  and  $\mathbf{c}$  describe the input-output relationship of the system.

$$\mathbf{A} = \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \quad \mathbf{c} = (c_1 \quad c_2)\tag{3-16}$$

### Spring-Mass Linear Oscillator

Since there are many types of oscillators available in continuous time, it seems reasonable to begin analysis of an oscillatory network by looking at physical systems. The differential equation of a simple, undamped spring-mass oscillator is described by

$$\ddot{\mathbf{x}} = -\omega^2 \mathbf{x}.\tag{3-17}$$

The system will oscillate at a continuous time frequency  $\omega$  indefinitely—one of the desired traits discussed above—when the initial condition is not at the equilibrium point. The second order system in Eq. 3-17 can be separated into two first order states by using

$$x_1 = x, \quad \dot{x}_1 = \dot{x}, \quad x_2 = \dot{x}_1, \quad \dot{x}_2 = \ddot{x},\tag{3-18}$$

and combined into state equations as

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.\tag{3-19}$$

Discrete time difference equations for the continuous time system can be developed when the inputs are piecewise continuous.<sup>48 pp. 830, 831</sup> These inputs arise from the sample and hold process and the resulting difference equations characterize the system at the sampling instants only, i.e., at  $t = 0, T, 2T, \dots$  where  $T \equiv$  Sampling Time.

The transfer function of a system of state equations is defined by<sup>49 p. 68</sup>

$$H(s) = \mathbf{c}[s\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}. \quad (3-20)$$

Let  $\Phi(s)$  be defined as the center portion of  $H(s)$  as

$$\Phi(s) = [s\mathbf{I} - \mathbf{A}]^{-1} = \begin{pmatrix} \frac{s}{s^2 + \omega^2} & \frac{1}{s^2 + \omega^2} \\ \frac{-\omega^2}{s^2 + \omega^2} & \frac{s}{s^2 + \omega^2} \end{pmatrix} \quad (3-21)$$

for the spring-mass system. The inverse Laplace transform  $\phi(T)$  is

$$\phi(T) = \begin{pmatrix} \cos(\omega T) & \frac{1}{\omega} \sin(\omega T) \\ -\omega \sin(\omega T) & \cos(\omega T) \end{pmatrix}, \quad (3-22)$$

giving the difference equations

$$\begin{pmatrix} x_1(n+1) \\ x_2(n+1) \end{pmatrix} = \begin{pmatrix} \cos(\omega T) & \frac{1}{\omega} \sin(\omega T) \\ -\omega \sin(\omega T) & \cos(\omega T) \end{pmatrix} \begin{pmatrix} x_1(n) \\ x_2(n) \end{pmatrix}. \quad (3-23)$$

These equations can be directly implemented in a linear two-PE network as shown in Figure 3-3.

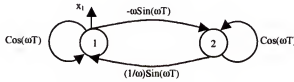


Figure 3-3 Original four-weight spring-mass oscillator network.

The input to the system is a unity impulse at time  $T = 0$ . The coefficients to the state equations are

$$\mathbf{A}_{SM} = \begin{pmatrix} \cos(\omega T) & \frac{1}{\omega} \sin(\omega T) \\ -\omega \sin(\omega T) & \cos(\omega T) \end{pmatrix} \quad \mathbf{b}_{SM} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \mathbf{c}_{SM} = (1 \quad 0). \quad (3-24)$$

It is seen that the state  $x_1$  represents the position of the mass, and  $x_2$  its velocity. A plot showing both states of this system with  $\omega = 2\pi 7$  rad/sec and  $T = 0.004609$  seconds (corresponding to a sampling frequency of 216.99 samples per second) is shown in Figure 3-4.

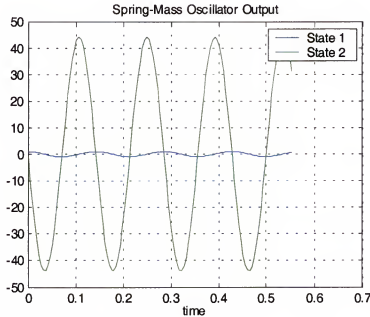


Figure 3-4 Spring-mass oscillator output.

### Conversion to Single-Weight Adaptation

The system is now a function of variables that can be controlled, that is  $\omega$  and  $T$ , as indicated in Eq. 3-23. The system is bound by one degree-of-freedom,  $\omega$ , given a fixed  $T$ . But a drawback to this configuration is that backpropagation treats each of the four weights numerically and independently, as illustrated in the  $A$  matrix in Eq. 3-15, rather than as a nonlinear function, making it, for all practical purposes a four-degree-of-freedom network. In other words it adapts  $W_{11} = \cos(\omega T) = a$  (where  $a$  is some real number), rather than  $W_{11} = f(\omega)$ , for instance. If, through backpropagation the proper ratio and nonlinear relationship of the four weights are not maintained, the oscillatory

nature of the system will be lost. In order to easily control the frequency of oscillation, and ensure that the response of the system is truly sinusoidal after adaptation, a conversion from four degrees-of-freedom to one is needed.

As a starting point, requiring that only one weight be adjustable, and knowing that the network must have at least one inhibiting connection, a concept for a more desirable response matrix may be

$$\mathbf{A}' = \begin{pmatrix} 0 & 1 \\ 1 & -\alpha \end{pmatrix} \quad \text{or} \quad \mathbf{A}' = \begin{pmatrix} 0 & -1 \\ 1 & \alpha \end{pmatrix}. \quad (3-25)$$

System theory<sup>49 p. 122</sup> allows matrices to be transformed from one to another by a similarity transformation matrix  $\mathbf{T}$ , where

$$\mathbf{A}' = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}, \quad \mathbf{b}' = \mathbf{T}^{-1}\mathbf{b}, \quad \mathbf{c}' = \mathbf{c}\mathbf{T} \quad (3-26)$$

*with  $\mathbf{T}$  such that  $\text{eig}(\mathbf{A}') = \text{eig}(\mathbf{A})$ .*

That is, a system defined using  $\mathbf{A}'$  is an equivalent realization of that containing  $\mathbf{A}$ , as long as the eigenvalues of both  $\mathbf{A}'$  and  $\mathbf{A}$  are the same. Going back to a generic form of  $\mathbf{A}$

$$\mathbf{A} = \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix}, \quad (3-27)$$

the eigenvalues are found by solving

$$\det(\mathbf{A}, \lambda) = \begin{vmatrix} W_{11} - \lambda & W_{12} \\ W_{21} & W_{22} - \lambda \end{vmatrix} = 0. \quad (3-28)$$

This reduces to

$$\lambda^2 - (W_{11} + W_{22})\lambda + W_{11}W_{22} - W_{12}W_{21} = 0, \quad (3-29)$$

and the eigenvalues are solved for by using the quadratic formula.

$$\lambda = \frac{(W_{11} + W_{22}) \pm \sqrt{(W_{11} + W_{22})^2 - 4(W_{11}W_{22} - W_{12}W_{21})}}{2}. \quad (3-30)$$

Similarly, assume  $\mathbf{A}'$  takes the second form shown in Eq. 3-25. The eigenvalues are found the same way as before, by solving

$$\det(\mathbf{A}' - \lambda' \mathbf{I}) = \begin{vmatrix} -\lambda' & -1 \\ 1 & \alpha - \lambda' \end{vmatrix} = 0. \quad (3-31)$$

This reduces to

$$\lambda'^2 - \alpha\lambda' + 1 = 0, \quad (3-32)$$

and the eigenvalues are solved for as

$$\lambda' = \frac{\alpha \pm \sqrt{\alpha^2 - 4}}{2}. \quad (3-33)$$

By setting Eq. 3-30 equal to 3-33, and using the values for  $W_{ij}$  given in Eq. 3-22,

$$\alpha = (W_{11} + W_{22}) = 2\cos(\omega T) \quad (3-34)$$

since

$$(W_{11}W_{22} - W_{12}W_{21}) = \cos^2(\omega T) + \sin^2(\omega T) = 1. \quad (3-35)$$

This implies that the  $\mathbf{A}'$  matrix is

$$\mathbf{A}' = \begin{pmatrix} 0 & -1 \\ 1 & 2\cos(\omega T) \end{pmatrix}. \quad (3-36)$$

A diagram of the final network is shown in Figure 3-5. Notice that one feedback loop has been eliminated, that there is a single inhibitive branch in the network, and that there is plasticity in only on weight. The frequency of oscillation can be initialized to any value desired, within the numerical limitations of the system.

The output of this network is shown in Figure 3-6, with  $\omega = 2\pi 7$  rad/sec,  $T = 0.004609$  seconds, and initial conditions  $x_1 = 1$ ,  $x_2 = -\cos(\omega T)$ . It is clear that the states have been transformed and no longer represent the position and velocity of the mass in a

spring-mass system. The physical meaning of the network has been clouded, sacrificed for numerical stability and simplicity.

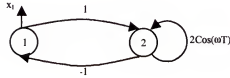


Figure 3-5 Single weight oscillator network.

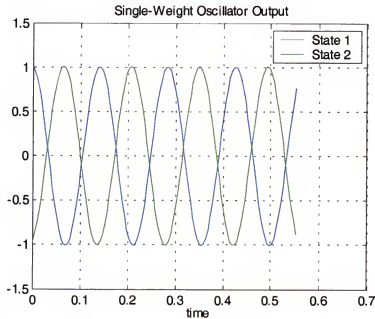


Figure 3-6 Single weight spring-mass oscillator output.

### Network Transfer Function Analysis

Now that the separate AR and MA models have been defined, and the specific configuration of the oscillators has been determined the transfer function for the model can be developed. Reviewing, the system consists of two modules. The first section of the topology, the oscillators, are made up of recursive circuits whose outputs are sine waves at fixed frequencies determined by the frequency-controlling weights. This



recursive weight's value is  $W_{\text{freq}} = 2\cos(\omega T)$ . The second, or output portion, are FIR filters whose function is to adjust the phase and gain of the input sine waves.

All inputs to the system are simply an impulse at time  $t = 0$ . These are scaled by the input weights  $W_{i1}$  and  $W_{i2}$ . Usually these are simplified further with  $W_{i1} = 0$  and  $W_{i2} = -\sin(\omega_0 T)$ . This combination results in a sine wave with zero phase at the frequency  $\omega_0$ . The transfer function of the system can be determined from the ratio of the z-transform of the outputs to that of the inputs

$$H = \frac{Y}{U}. \quad (3-37)$$

A block diagram for a simple single-mode system is shown in Figure 3-7. The impulse response  $h$  can be separated into two components,  $h_R$ , the impulse response of the recursive portions of the network, and  $h_F$ , the impulse response of the FIR filter portions.

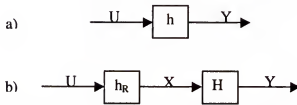


Figure 3-7 Impulse response of total system and two part system.

The total impulse response is

$$h = h_R * h_F, \quad (3-38)$$

and the transfer function is

$$H = H_R H_F. \quad (3-39)$$

An entire single channel system made up of  $M$  modes is shown in Figure 3-8.

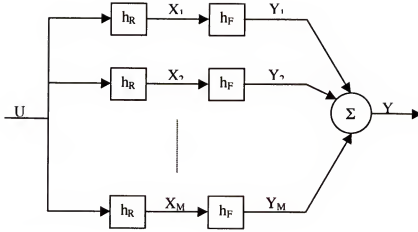


Figure 3-8 Multi-modal impulse response.

Since the impulse responses for each mode are in parallel, the transfer function for the entire system can easily be calculated as a sum of the transfer function of each oscillator-filter pair.

$$H_T = H_1 + H_2 + \cdots + H_M, \quad (3-40)$$

$$H_T = H_{R1}H_{F1} + H_{R2}H_{F2} + \cdots + H_{RM}H_{FM}. \quad (3-41)$$

The transfer function for a single mode in the recursive network is developed first and then expanded to include all modes in the channel. The recursive oscillator has two nodes with states  $x_1$  and  $x_2$ . Analysis begins with difference equations for each of these states.

$$x_1(n) = W_{i1}u(n) - x_2(n-1), \quad (3-42)$$

$$x_2(n) = W_{i2}u(n) + 2\cos(\omega T)x_2(n-1) + x_1(n-1). \quad (3-43)$$

Next z-transform of both sides of the equation is taken,

$$X_1(z) = W_{i1}U(z) - X_2(z)z^{-1}, \quad (3-44)$$

$$X_2(z) = W_{i2}U(z) + 2\cos(\omega T)X_2(z)z^{-1} + X_1(z)z^{-1}. \quad (3-45)$$

The reference to X and U as a function of z will be implied for convenience. From Eq. 3-45  $X_2$  can be solved for as

$$X_2 = \frac{W_{i2}U + X_1z^{-1}}{1 - 2\cos(\omega T)z^{-1}}. \quad (3-46)$$

Substituting Eq. 3-45 into Eq. 3-44  $X_1$  is solved for as

$$X_1 = \frac{W_{i1}U - (2\cos(\omega T)W_{i1}U + W_{i2}U)z^{-1}}{1 - 2\cos(\omega T)z^{-1} + z^{-2}}. \quad (3-47)$$

The transfer function for the recursive network is

$$H_R = \frac{X_1}{U} = \frac{W_{i1} - (2\cos(\omega T)W_{i1} + W_{i2})z^{-1}}{1 - 2\cos(\omega T)z^{-1} + z^{-2}}. \quad (3-48)$$

The transfer function for a single mode FIR filter is easily developed from its difference equation

$$y(n) = W_0x_1(n) + W_1x_1(n-1) + \dots + W_Nx_1(n-N), \quad (3-49)$$

where N is the order of the filter.

The transfer function is found by taking the z-transform of both sides of Eq. 3-49 and dividing through by the input, in this case  $X_1(z)$

$$H_F = W_0 + W_1z^{-1} + \dots + W_Nz^{-N}. \quad (3-50)$$

The complete frequency response for one mode is

$$H = H_R H_F = \frac{(W_{i1} - (2\cos(\omega T)W_{i1} + W_{i2})z^{-1})(W_0 + W_1z^{-1} + \dots + W_Nz^{-N})}{(1 - 2\cos(\omega T)z^{-1} + z^{-2})}. \quad (3-51)$$

This is clearly an ARMA process, and is the product of AR and MA systems. Pairs of poles are located on the unit circle at angles corresponding to  $\pm$  the digital frequency of the oscillator. The transfer function for an entire M-mode channel is

$$H_T = \sum_{m=1}^M \frac{(W_{i1m} - (2\cos(\omega_m T)W_{i1m} + W_{i2m})z^{-1})(W_{0m} + W_{1m}z^{-1} + \dots + W_{Nm}z^{-N})}{(1 - 2\cos(\omega_m T)z^{-1} + z^{-2})} \quad (3-52)$$

with pairs of poles for each frequency located around the unit circle. A simplification can be realized if all the oscillators are set to provide zero-phase sine waves, making the input weights for node 1 in all the oscillators zero. So the transfer function reduces to

$$H_T = \sum_{m=1}^M \frac{W_{l2m} z^{-1} (W_{0m} + W_{1m} z^{-1} + \dots + W_{Nm} z^{-N})}{(1 - 2 \cos(\omega_m T) z^{-1} + z^{-2})}. \quad (3-53)$$

There are complex conjugate pairs of poles on the unit circle for each digital frequency, a zero at the origin, and zeros corresponding to the FIR characteristic equation, depending on the filter order and phases of each frequency.

### Linear Damping

An enhancement to the feedforward system allowing the system to model damping, can be realized by retraining the network to follow a signal as it changes over time. By adding a bias term a shift from zero-mean can now be modeled as well. Let the output of a single mode adaptive system be

$$y(t) = a \sin(\omega t + \phi) + b. \quad (3-54)$$

Analysis will be performed for a single mode with the understanding that the results can be expanded to include any number of modes. Suppose the desired response  $d(t)$  has a smoothly increasing amplitude  $A$  and frequency and phase  $\omega$  and  $\phi$  that tend to wander as the amplitude increases and decreases. If the system  $y$  is trained to follow the entire signal  $d(t)$ , the best that can be accomplished is an averaging solution over the entire time-span of the signal  $d(t)$ . But now retrain the system  $y$  at regular time intervals  $t = T, t = 2T$ , and so forth, using the trained results of the previous network as initial conditions for the next. If the duration of the signal  $d(t)$  is divided into  $N$  segments  $T$  wide, the system can be adapted to model amplitude, frequency, phase and bias as they change over time, i.e., a LLM system, so Eq. 3-54 can be rewritten as

$$y_{nT}(t) = a_{nT} \sin(\omega_{nT}t + \phi_{nT}) + b_{nT} \quad nT < t < (n+1)T, \text{ for } n = 0, \dots, N-1. \quad (3-55)$$

This is an improvement on the first system since not only can modal changes be tracked, but damping in the signal is now modeled in a linear sense. A network adapted to each window of data  $T$  seconds wide will now model the average behavior of that window. As  $T$  becomes small, this approximation approaches a smooth damping curve. This is an important step towards modeling the response of a nonlinear oscillatory system such as aircraft flutter and limit cycle oscillation. Of course practical tradeoffs must be made between training speed, window size and network accuracy.

This data windowing technique suggests an LLM system that uses signal segmentation and model training and switching to continually synthesize a flight response.

### Local Linear Models

System identification and time series prediction are examples of how engineers approach the time-honored problem of function approximation. The traditional starting approach is to use statistical methods based on linear models.<sup>50</sup> Dynamic modeling is the next generation of techniques available to engineers when the complexity of a time series is linked to a high order nonlinear dynamical system.<sup>51</sup> A dynamically modeled system must either be nonlinear or linear with time varying parameters. Nonlinear dynamical systems have been successfully modeled using ANNs, but if a system operates in multiple modes and dynamics are drifting or switching, standard fixed-parameter models are likely to fail to represent the underlying input-output relationship. Moreover, they don't reveal the dynamical structure of the system. But a family of linear models matching local portions a problem space can potentially represent such a system.<sup>52</sup>

The capability of LLMs to follow a complex time series was illustrated in a time series prediction contest held at the Santa Fe Institute in 1991.<sup>5</sup> The winner of the competition used a neural network architecture, while the results of the second place entry, using a local linear model, were nearly as good. A second competition was held in Leuven, Belgium in 1998<sup>53</sup> to assess the changes and improvements that had occurred in the field of time series prediction since the Santa Fe competition. Entrants were given a time series consisting of 2,000 points and were asked to predict the following 200 points. LLMs generated the top two entries and only LLMs were able to predict the first 80 points accurately. The strong showing in both competitions supports the claim that LLMs are very appropriate for time series prediction. The tools for developing LLMs are well documented by Principe<sup>52</sup> et al. and so will only be briefly reviewed here. The goal is to exploit the positive aspects of LLM form modeling flutter and LCO wing responses, providing insight into their modal behavior.

The basic assumption underlying the use of learning systems for modeling purposes is that the behavior of the system can be described in terms of a training set consisting of sampled observations of the dynamical system

$$\mathbf{y} = [y(n), y(n - \tau), \dots, y(n - (N - 1)\tau)], \quad (3-56)$$

where  $\tau$  represents a normalized delay for a given input vector

$$\mathbf{x} = [x(n), x(n - \tau), \dots, x(n - (M - 1)\tau)]. \quad (3-57)$$

For simplicity assume  $\tau = 1$ . Dynamic modeling of this system takes the form

$$y(n) = F(\mathbf{x}, \mathbf{y}). \quad (3-58)$$

for a sufficient<sup>54</sup> training data size  $N$  and an input vector of length  $M$ . Equation 3-58

describes an ARMA system. A much more desirable model would be one that is strictly FIR, in other words

$$y(n) = F(\mathbf{x}). \quad (3-59)$$

This would ease learning and reduce the size of the architecture of any neural models used.

If  $F$  were linear, an approximation  $\tilde{F}$  could be described and parameterized globally in a number of ways. However if  $F$  is known or assumed to be nonlinear, as is the case in most real dynamical systems, defining  $\tilde{F}$  globally becomes much more difficult, and LLMs become an attractive alternative. A family of such models can be defined such that

$$\tilde{F}(\mathbf{x}(n)) = \bigcup_{r=1, \dots, R} \tilde{F}_r(\mathbf{x}(n)). \quad (3-60)$$

where  $R$  is region over which the LLMs are defined.

#### **Generalized Likelihood Ratio Test Basis for Detecting Signal Changes**

The LLM method proposed in this work models a large system as a set of small, linear subsystems in a piece-wise manor, switching between known models as needed, with an ever-growing database of models re-adapted to subsystems over time as changes in the response warrant.<sup>55</sup> In order to implement such a process, a method is needed to detect when a particular model is no longer adequate and identify the proper time for switching to a different model, or for updating the system through adaptation. Equally important is a method that can correctly identify the proper model to be used after a change, or identify the need and proper initial conditions for retraining. Whether the desired response is deterministic or stochastic, the change from one model to the next can itself be considered random. Therefore stochastic methods can be exploited as a device to detect changes in the process as they occur.

A detection statistic is a numerical value that is calculated as the system runs and is based on the behavior of the system and its representative model outputs. The statistic

can be observed for change, providing a measurement of the current model' s capability to represent the desired response, and giving an indication that switching or retraining is necessary. Switching or retraining is accomplished when the statistic goes beyond a preset threshold.<sup>12 p. 118</sup> The same statistic can be used to identify the proper model to select after the change. Since this work is based on training ANNs using the MSE as a cost function, it makes sense that MSE be the detection statistic of choice.

In 1982 Appel and Brandt<sup>56</sup> published an approach using a Neyman-Pearson test, called the Generalized Likelihood Ratio Test for deciding between two hypotheses: the null hypothesis,  $H_0$ , that no change occurred within a window of  $N$  samples, and the posited hypothesis,  $H_1$ , that a change did occur at some change point at time  $T$ . The time line and parameters for these hypotheses of the three model GLRT are shown in Figure 3-9.

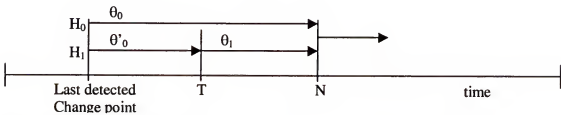


Figure 3-9 Three model GLRT.

The change detection method in this work is based on the GLRT since it is not constrained by any single error distribution. It is flexible enough, at least in theory, to handle complex data sets as long as their probability density functions can be defined, or a reasonable approximation made. The GLRT will be modified to use only one active model in a given window, as will be explained. Other methods exist for detecting changes in the state of a system, such as Hidden Markov models, Analysis of Variances, as well as Fourier Transform and Autocorrelation techniques. The advantage of GLRT is



that it does not require a global search for a potential change point, and can be simplified to exploit the Gaussian nature of error used to train and evaluate ANNs. All models in a given set can be evaluated simultaneously, a winner selected, or if necessary an optimal set of initial conditions for training new models can be identified.

The work for which GLRT is being considered has a desired response that cannot be considered stationary, but can be approximated by a sum of sinusoidal terms over small windows of that response. For example, consider a system whose output over a fixed window of time  $T$  is consistent, and can be represented by a sum of sinusoids

$$d(t) = \sum_{m=1}^M A_m \sin(\omega_m t + \phi_m) + b_m, \quad (3-61)$$

with amplitude  $A$ , frequency  $\omega$ , phase  $\phi$ , and perhaps an offset bias  $b$ . These parameters may be time-varying, but can be considered constant for a small time-window,  $t_1 < t < t_2$ ,  $t_2 = T + t_1$ . Further, suppose that the frequencies and phases stay nearly constant throughout the entire signal duration. The amplitudes of the signal are therefore the most significantly changing feature. The system can be considered stationary for that time period. Fancourt's discussion<sup>12, page 109</sup> of the concepts of stationary, piece-wise stationary and wide-sense stationary are excellent. His initial definition of a stationary random process—a process whose local statistics are invariant with respect to the time the process has been running, assuming all transients have died out—is sufficient for this work.

The generalized log-likelihood ratio for a memoryless process is defined by Fancourt as

$$L(T) \Big|_{\hat{\theta}_0, \hat{\theta}_1} = \sum_{m=1}^{T-1} \ln \left[ \frac{P(x(m); \hat{\theta}_1)}{P(x(m); \hat{\theta}_0)} \right] + \sum_{m=T}^n \ln \left[ \frac{P(x(m); \hat{\theta}_1)}{P(x(m); \hat{\theta}_0)} \right]. \quad (3-62)$$

The parameters in the equation,  $\hat{\theta}_0, \hat{\theta}_0', \text{ and } \hat{\theta}_1$  are approximations to the actual parameters  $\theta_0, \theta_0', \text{ and } \theta_1$  shown in Figure 3-9, which of course are unknown. The ratio is slightly different for a system with memory. Using Bayes' theorem the likelihood must be evaluated recursively as

$$L(T) \big|_{\hat{\theta}_0, \hat{\theta}_0', \hat{\theta}_1} = \sum_{m=1}^{T-1} \ln \left[ \frac{P(x(m) | X(m-1))}{P(x(m) | X(m-1))} \right] + \sum_{m=T}^n \ln \left[ \frac{P(x(m) | X(m-1))}{P(x(m) | X(m-1))} \right]. \quad (3-63)$$

That is, the probabilities are now conditional on past data. Beginning with this form of the ratio, Fancourt introduces an example that is applicable to this work. Specifically, consider a change in a process with memory using a parametric predictive model under the assumption that the errors are Gaussian with variance  $\sigma^2$ . Fancourt postulates that if the system has a finite effective memory,  $L(T)$  can be shown to reduce to

$$2L(T) \big|_{\hat{\theta}_0, \hat{\theta}_0', \hat{\theta}_1} = (T-1) \ln \left[ \frac{\sigma_0^2}{\sigma_0'^2} \right] + (N-T+1) \ln \left[ \frac{\sigma_0^2}{\sigma_1^2} \right]. \quad (3-64)$$

This is the Three Model form of the solution. Assuming Normal distributions, the Maximum Likelihood (ML) estimates of the parameters  $\sigma_0, \sigma_0', \sigma_1$  and the mean  $m$  are

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N (\varepsilon(n) - E[\varepsilon(n)])^2, \quad m = E[\varepsilon(n)] = \frac{1}{N} \sum_{n=1}^N \varepsilon(n). \quad (3-65)$$

The variance for  $\hat{\theta}_0, \sigma_0^2$  is calculated over the entire window, since it assumes the window contains no model change. The variances  $\sigma_0'^2$  and  $\sigma_1^2$  are calculated from the data before the assumed change time  $T$  and after  $T$ , respectively.

Strictly speaking, the variances used are variances on model parameters for the three models in the ratio. These model parameters may not be Independently Identically Distributed (iid). The errors between the model output and the desired response can be

consider iid, however (in fact they must be considered iid in order to use MSE as the criterion for model training).

The algorithm proceeds as follows. First a window of data from the desired response with fixed length  $N$  is selected. It is assumed that a database of models exists with a minimum of size of one corresponding to a model matching the first portion of the data in the window. The model output, in this case the output of an ANN, is compared the desired response  $d(n)$  in the window and an error is defined as

$$\varepsilon(n) = d(n) - y(n) \quad (3-66)$$

where  $y(n)$  is the output of the ANN at sample  $n$  for  $n = 1, \dots, N$ . This is simply the same error that is normally calculated for training any ANN. The variance  $\sigma_0^2$  is calculated once using the entire window of data, since it assumes no change is contained in the window. The other two variances,  $\sigma_0'^2$  and  $\sigma_1'^2$  are calculated for all times  $t = T$  over the entire window. They are the result of the posited hypothesis that a change does exist in the data window. In other words, the value  $T$  is moved from one end of the window to the other, the variances are continually calculating which are used to calculate the value of  $L(T)$  as  $T$  moves. The result is a ratio value for each time  $T$  in the window. This is accomplished for every model in the model database. As can be seen in Eq. 3-64, when the model for  $d(n)$  matches, the ratio  $L$  increases rapidly with every sample  $n$ . This continues until such time  $T$  that a change in the response occurs. The  $L$ -values for the model originally matched to the signal will now drop to very small values, indicating that a change has indeed occurred to the signal. The time  $T$  where  $L$  peaks is identified as the change point.

The final step in the model detection process is to identify the correct model for use after the change, or notify the training algorithm that a new model is required if no

current model exists. The post-change model is identified by comparing MSE values for each model in the database after the change time. If a model's MSE is lower than the criteria it is declared the winner and becomes the new model for the current data from the change point forward. If no model's MSE passes the criteria, the closest model to the minimum is selected as the initial condition for training a new model to match the data window starting at the change point. For training, it must again be assumed that no change in the data occurs over a minimum fixed length of time. Once a new model is trained it is added to the database and the process continues with further data windows until the entire data sequence has been evaluated and modeled.

#### **A Proposed Modification to the GLRT**

Training and maintaining three models for each window can be very expensive. In an effort to reduce the number of calculations necessary for using the GLRT a modification to Fancourt's original work is proposed by making an assumption that with fixed inputs to a MA adaptive model in a given window of time, such as the modal sinusoids of the AR components, that there will be a one-to-one mapping of those inputs to an optimal MA model. This mapping would therefore result in errors that are smaller than errors attributed to any other model. This is an apparent violation of the concept of an insufficient statistic,<sup>57</sup> that is, that there is a many-to-one mapping from different stationary regimes to identical statistics of the parts of the signal that cannot be explained by the model. But for this work, where signal change detection is based on an acceptable MSE, it can be argued that for given fixed inputs gradient descent training will always result in the same FIR filter mapping when the filters are initialized the same way for each trial. Experiments in the next chapter will bear this out. When the oscillators are set at the flutter frequencies and FIR filters initialized at the free vibration modal value with

zero phase filter—all values made available from linear flutter analysis—adaptation quickly results in consistent models. In other words a sufficient statistic is found. The GLRT simplification takes advantage of this as well as the supposition that a good model exists for the data at the beginning of the sequence and simplifies the algorithm as follows.

Suppose that instead of the three models existing for the data in the window shown in Figure 3-9 only one model existed, corresponding to  $\hat{\theta}_0'$ , for the first few samples of the window. The ML estimate of the variance of its modeling errors,  $\hat{\sigma}_0'^2$ , can be calculated (from this point forward all values given for a variance  $\sigma^2$  will reflect the ML estimate of the variance of a model output with respect to a desired response as discussed above). For each value of T an approximation to the other two variances,  $\hat{\sigma}_1^2$  and  $\hat{\sigma}_0^2$ , are calculated using the single model for outputs for  $n = 1 - N$  for  $\hat{\sigma}_0^2$  and  $n = T - N$  for  $\hat{\sigma}_1^2$ . These two variances effectively measure the single model's ability to match the entire data window and the portion of data from the current time T to the end of the window respectively. Since the single model may no longer be optimal these errors may not be iid, but the estimates will serve to identify a 'good' or 'bad' fit. In other words, if the single model is not matched to the portion of data compared, the variance of its errors will be larger than that of the portion of data where the model is applicable.

Two results are immediately apparent from these simplifications. First, if the model is sufficient for the entire data window, all three variances will be approximately the same (within a given threshold) and the L-values will remain low. Second, if a signal change is indeed present at some time in the data window, the variances  $\hat{\sigma}_1^2$  and  $\hat{\sigma}_0^2$  will be larger than  $\hat{\sigma}_0'^2$  simply because the model won't match as well at those points. These

results and postulations indicate that the proposed simplifications can be used to streamline the process by substituting the defined variances into Eq. 3-64. Equation 3-65 shows the ML estimate for a variance. The relationship between MSE and the variance when the mean is nonzero can be shown to be

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N \varepsilon^2(n) - \frac{2}{N} \sum_{n=1}^N \varepsilon(n) \frac{1}{N} \sum_{n=1}^N \varepsilon(n) + \frac{1}{N} \sum_{n=1}^N \left( \frac{1}{N} \sum_{n=1}^N \varepsilon(n) \right)^2, \quad (3-67)$$

$$\sigma^2 = \text{MSE} - m^2. \quad (3-68)$$

It is asserted that the MSE for the known model is less than or equal to the other models. That is

$$\text{MSE}_0' \leq \text{MSE}_0; \quad \text{MSE}_0 \leq \text{MSE}_1. \quad (3-69)$$

The variance  $\hat{\sigma}_0'^2 = \text{MSE}_0'$  since  $\hat{\theta}_0'$  is assumed to be zero-mean. But since the other models are not optimal their means may be nonzero. Therefore, from Eq. 3-67 it can be said that

$$\begin{aligned} \hat{\sigma}_0'^2 &\leq \hat{\sigma}_0^2 + \hat{m}_0^2; \\ \hat{\sigma}_0^2 + \hat{m}_0^2 &\leq \hat{\sigma}_1^2 + \hat{m}_1^2. \end{aligned} \quad (3-70)$$

These are the limitations that must remain consistent for the approximation to be accurate. Equation 3-70 is interpreted to say that any nonzero mean for the unknown models increases the sensitivity of the algorithm to signal changes. This relaxes the zero mean assumption on all but the original known model.

In order to further evaluate the ramifications of this simplification, define two sub-L-values from the two portions of Eq. 3-64,  $L_1$  and  $L_2$  as

$$L_1 = (T-1) \ln \left[ \frac{\hat{\sigma}_0^2}{\sigma_0^2} \right], \quad \hat{L}_1 = (T-1) \ln \left[ \frac{\hat{\sigma}_0'^2}{\sigma_0'^2} \right], \quad (3-71)$$

$$L_2 = (N - T + 1) \ln \left[ \frac{\sigma_0^2}{\sigma_1^2} \right] \quad \hat{L}_2 = (N - T + 1) \ln \left[ \frac{\hat{\sigma}_0^2}{\hat{\sigma}_1^2} \right]. \quad (3-72)$$

By observing and comparing the behavior of these sub-L-values it can be determined whether or not the proposed substitutions are reasonable and will result in the positive identification of a signal change in a manner similar to the true GLRT. A simple exercise on these values is done for four cases. Only the relative magnitudes of the variances are necessary to illustrate the behavior of each sub-L-value. It is assumed that one model exists for each case and that it fits the data at the beginning of the window. The first case examines the values when a change is indeed present in the window, at a time prior to the change. The second is similar, looking at the values at the time of the change. Third, the values are examined after the change time. The final study looks at the behavior of the sub-L-values when there is no signal change in the window.

In the first case, variance  $\sigma_0^2$  will be less than or equal to the MSE value used as a training criteria since the original model matches the data before the signal change, and is a small value. The variances  $\sigma_0^2$  and  $\sigma_1^2$  will be much larger;  $\sigma_0^2$  because it models the entire data window, including the signal change and therefore won't achieve a very good minimum MSE, and  $\sigma_1^2$  because it also models across a signal change, and but perhaps small since it matches the data after the change. The variance  $\hat{\sigma}_0^2$  will also be larger, since it is calculated with a modeled matched to the data prior to that change and calculated in a window with a change. The same holds for  $\hat{\sigma}_1^2$ . So for the first case the sub-L-values would appear as

$$\begin{aligned} L_1 &> 0, \\ L_2 &> 0, \\ \hat{L}_1 &> 0, \end{aligned}$$

$$\hat{L}_2 \approx 0.$$

The actual values for  $L_1$  and  $\hat{L}_1$  will not be the same, but they both are greater than zero and will continue to increase as time approaches the change time  $T$ .

In the second case, when time is the change time  $T$ , the variance  $\sigma_0'^2$  will again be less than or equal to the MSE value used as a training criteria since the original model matches the data up to  $T$ , and is a small value. The variance  $\sigma_0^2$  will remain large because its model spans the change, but  $\sigma_1^2$  will now be small since it is the result of modeling the data past the break to the end of the window. The variance  $\hat{\sigma}_0^2$  will again be large as will  $\hat{\sigma}_1^2$  for the same reasons as case 1. The case 2 sub-L-values would appear as

$$\begin{aligned} L_1 &> 0, \\ L_2 &\approx 0, \\ \hat{L}_1 &> 0, \\ \hat{L}_2 &\approx 0. \end{aligned}$$

Again, actual values will not be the same, but they are greater than zero and in all cases are at their maximum value since  $\sigma_0'^2$  is at its optimum.

In the third case, when time has passed the change time  $T$ , the variance  $\sigma_0'^2$  will begin to increase since it now spans a signal change. The variance  $\sigma_0^2$  will remain large because its model also spans the change, but  $\sigma_1^2$  will now be small since it is the result of modeling the data past the break to the end of the window. The variance  $\hat{\sigma}_0^2$  will again be large as will  $\hat{\sigma}_1^2$  for the same reasons as case 1. The case 3 sub-L-values would appear as



$$L_1 \approx 0,$$

$$L_2 > 0,$$

$$\hat{L}_1 \approx 0,$$

$$\hat{L}_2 \approx 0.$$

This case shows that the L-values for the true GLRT may not decrease to zero since the after-change model matches the data. The model-reduction method may be the more optimal method since its values appear to drop rapidly after a signal change.

The final case shows what happens when there is no signal change contained in the data window. The variance  $\sigma_0^2$  will be small as will  $\sigma_0^2$  and  $\sigma_1^2$ , since they all match the same data set. The same holds true for  $\hat{\sigma}_0^2$  and  $\hat{\sigma}_1^2$ . The case 4 sub-L-values would appear as

$$L_1 \approx 0,$$

$$L_2 \approx 0,$$

$$\hat{L}_1 \approx 0,$$

$$\hat{L}_2 \approx 0.$$

This case shows that the L-values for the true GLRT and the model-reduction method all remain very low over the entire data window, as they should.

This simple evaluation of the model-reduction modification to the GLRT shows that in all possible instances the approximation behaves nearly the same way as the original algorithm and should therefore be a viable option for signal change detection. Extensive experimentation will be shown in Chapter 4 that illustrates the success of the proposed GLRT modification.

### **GLRT Practical Considerations**

Admittedly there have been some liberties taken with the theory of the GLRT, such as the assumption that all the errors can be considered white noise, and that a good initial model always exists at the beginning of each data window. Since these assumptions are

not always true, there are some practical considerations that must be recognized when employing the GLRT, whether modified or not. The assumption that distribution of the noise is Gaussian isn't particularly detrimental, though. Consider for a moment the consequences if it is not. The ML estimates of the variances will be incorrect, but their numerical values will be higher than those of properly trained modeling errors. The effect on the results of the GLRT evaluation won't change. That is, if a signal change does exist it will still be identified whether or not the errors are truly Gaussian.

The proper threshold to use when trying to detect a change in a model is sometimes difficult to determine. When a truly matching model exists for the data in the window, the L values become large quite rapidly, and are much greater than those calculated for the other models in the database. The threshold is easy to set in this case since a large value can be selected rather arbitrarily. But when none of the database models are exact the L-values tend to remain small and irregular especially near either end of the data window. This can easily be the case when evaluating data such as the Test line 17 series, where it is known a priori that the signals are constantly changing. Thresholds that are set too low will result in false alarms. The GLRT technique is therefore only 'semiautomatic,' rather than fully self controlling since some manual oversight of the process is required when questionable models exist.

In this work it is assumed that the first model to break the threshold is the proper model, since its L values would increase the fastest. Therefore the change time (or switch time as it will sometimes be referred to) will be determined by the peak of that slope.

A switch point that exists very near the beginning of a data window is very difficult to detect because the statistics are not as robust since they are based on a smaller number

of samples. A practical minimum limit to the data must be set where it is assumed that no change has occurred in order for the  $L$ -values to be stable. In this work that value was determined to be about 15 samples.

The GLRT statistics are based on a Boolean-type hypothesis. It is logical, then, that the method fails if the data window being evaluated contains more than one switch point. The process by which this problem is avoided is simple. Begin with as small a window of data as is practical. It must be assumed that it contains no switch points. This can easily be verified by a quick comparison of MSE values. One by one add additional data points to the window until a switch point is detected and a new model identified, either from the existing database or from training a new one. The practical starting point for training a new model is to begin with the model closest to the correct solution, as measured by MSE. This close-model can then be fine-tuned to match the data from the switch point on. This is the practice followed throughout this work.

### **Chapter Summary: The Initial Model**

The model paradigm that has been developed in this chapter follows that proposed in Chapter one. Local linear models in the form of modular ANNs are designed using the first 16 modes identified in linear flutter analysis as initial conditions for the recursive linear oscillators. The output of each oscillator is modified for phase and gain by an adaptive linear FIR filter. These models are trained to match portions of accelerometer signals either identified explicitly through post-flight analysis of the data, or by means of simplified GLRT as an algorithm to test the model currently in use and identify the need for change to that model. The input to the model is simply a vector of impulses at time zero. The oscillatory network modules are autonomous, needing no further stimulus. The initial difference equations of the LMM is

$$\begin{aligned}
x_m(n) = & W_{i1m}\delta(n) - W_{i1m}\delta(n-1)2\cos(\omega_m T) - W_{i2m}\delta(n-1) \\
& + 2\cos(\omega_m T)x_m(n-1) - x_m(n-2) \quad ,
\end{aligned}
\tag{3-73}$$

$$y(n) = \sum_{m=1}^{16} W_{0m}x_m(n) + \dots + W_{pm}x_m(n-p) ,$$

where  $\delta(n)$  is a discrete-time impulse,  $W_{i1m}$  and  $W_{i2m}$  are first and second oscillator input weights for the  $m^{\text{th}}$  mode, and  $W_{0m}, \dots, W_{pm}$  are FIR weights for  $p$ -order filters. The oscillator outputs for each mode at sample  $n$  is  $x_m(n)$ , and the system output at sample  $n$  is  $y(n)$ . This clearly represents a sequential system with a SIMO AR model of the physical structure feeding a MISO MA system to completely simulate the desired response. The details of the model, including filter orders and training schemes will be developed through experimentation in the next chapter.

## CHAPTER 4

### MODEL TUNING AND REFINEMENTS

This chapter gives experimental results that are used to tune and refine the paradigm proposed in Chapter 1 for modeling an accelerometer signal. Chapter 3 established and defined the two major paradigm parts: the local linear models to synthesize segments of accelerometer data; and the simplified GLRT for signal segmentation. The LLMs consist of two modules. First a parallel set of recursive networks is used to properly place the poles of the system, behaving as a hybrid physical model of the structure in the form of oscillatory signal generators operating at the expected flutter frequency, establishing the modes of vibration. Each oscillator output becomes the input to an adaptive FIR filter used to adjust the phase and gain of each mode, performing the role of adjusting each mode to the affects of an aerodynamic forcing function. A model is trained and its parameters remain fixed until the signal changes sufficiently to warrant a shift in the model. The change is detected by the second part of the paradigm, the reduced-model simplified GLRT.

Before the paradigm can be used for practical data analysis there are some considerations for each piece of the process that must be addressed. The limitations and training parameters of the linear oscillators need to be established. The parameters for using simplified GLRT to detect signal changes need to be defined. It is known that simplified GLRT requires an assumption that a signal is time-invariant for a period of time sufficient to establish beginning statistics. What are the limits of this period? And how rapidly can a signal change given the data rates available from the accelerometers?

Can simplified GLRT detect changes in signals with more than one mode? Consideration must be given to the practical limits of the LLMs. It is known that LCO signals are nonlinear, and since this process seeks to model them with linear models for a period of time, the limits of the time constraints need to be measured. Also, the training parameters for the LLM as a complete set need to be established.

### **Chapter Organization**

This chapter seeks to address these issues through a series of experiments focusing on each part of the process. The LLMs will be evaluated first, including the initialization of the oscillators and how best to train them and their corresponding FIR filters. The oscillator architecture will be evaluated, as well as that of the FIR filters. Backpropagation methods will be determined and parameters set. Evaluation of the ability and limitations of frequency adjustment will be analyzed. Some examples will be given illustrating the ability of the oscillator to train to a single frequency and to a bi-modal signal, once the FIR filters have been attached. Finally a slight modification to the oscillator networks will be introduced that change the two nodes from strictly linear to saturated linear nodes, limiting the output to just beyond  $\pm 1$ .

Next, the practical use and limitations of simplified GLRT is evaluated by a sequence of experiments designed to look at different rates of signal change, bi-modal signals, and continuously changing signals. These experiments are designed to set the limits and illustrate the results that can be expected from the simplified GLRT when applied to signals that are similar to flight test accelerometer data. The bi-modal signal used in later experiments is an example of a simulated increasing LCO response containing energy from two discrete modes, for instance. These experiments will show

what can be expected of the L-values produced by the algorithm and how best to exploit the information provided by them.

The last section of the chapter will look at the actual accelerometer data in order to fine-tune model parameters, establish training guidelines and methods, FIR filter order and final LLM definition. Analysis of the realistic segment length is performed in order to get an idea of how well (or how poorly) the LLM will perform, especially as the segment length increases. The signals will be segmented manually, with increasing signal lengths. Results of these experiments will provide the final building blocks of the LLM system for LCO data synthesis.

### **Linear Oscillator Experiments**

#### **Network Training**

The intended use of the oscillator network is to provide sinusoidal inputs at several frequencies to banks of adaptable FIR filters. Since free vibration analysis of the aircraft components provides a set of expected frequencies, it follows that this information should be used to set initial conditions of the oscillators. This lets the oscillators operate initially at a frequency in the neighborhood of those expected in the training data, rather than simply using random weight initialization. This reduces the size of the solution space that must be searched by a considerable amount.

The two-node linear neural network is shown again in Figure 4-1. The frequency of oscillation is controlled by the adaptation of a single weight, from whose value the final frequency of oscillation can be directly determined.

The weight is initialized so that the network oscillates at a given frequency, by setting

$$W_{freq} = 2\cos(\omega_i T), \quad (4-1)$$

where  $\omega_0$  is the initial continuous-time frequency, and  $T$  is the sample time. A complete diagram of the network's architecture is given in Figure 4-2. Nodes 1 and 2 are recursively connected with a weight and delay of one sample. The inputs to each node are scaled by input weights  $W_{i11}$  and  $W_{i22}$ . Node 2 has a delayed feedback loop to itself with a one-sample delay.  $W_{22}$  is the frequency controlling weight,  $W_{freq}$ , and is adjusted during adaptation to tune the network to any desired frequency within the limitations sample rate. The weights coupling nodes 1 and 2 have fixed values

$$W_{21} = 1, \quad W_{12} = -1, \quad W_{11} = 1. \quad (4-2)$$

Input weights can be adjusted as a pair to begin the oscillation at any arbitrary phase. The weight for the first input can be set at any phase angle  $\theta$ , with the value for the second weight determined by the angle  $\theta$  plus one time-step at the current frequency as

$$\begin{aligned} W_{i11} &= \sin(\theta) \\ W_{i22} &= -\sin(\theta + \omega_i T) \end{aligned} \quad (4-3)$$

There are two alternatives for adapting the input weights during training. The first is to train the node-1 input weight along with the frequency weight, setting the second input weight as a function of the first. Alternatively the input weights can remain fixed through backpropagation and adjusted after each epoch depending on the latest value of  $\omega_0$ . Trial and error with both determined the second method to be the more practical and stable alternative.

The limits of  $W_{freq}$  are defined by

$$-2 < W_{freq} < 2, \quad (4-4)$$

with  $-2$  corresponding to a digital frequency of  $\pi$  rad/sec (or in other words half the sample frequency), and  $2$  to dc. Since the PE's are linear, the actual numerical value for



$W_{freq}$  found through backpropagation can fall well outside these bounds, resulting in stability problems for the network. Further, the numerical differences between two frequencies can be very small. For instance, at the sample rate of 216.99 Hz,

$$W_{freq\ 7.0} = 1.959056 \Rightarrow 7.0\text{ Hz},$$

$$W_{freq\ 7.2} = 1.956692 \Rightarrow 7.2\text{ Hz},$$

$$\Delta W_{freq} = 0.002364\text{ Hz}.$$

A change in weight values of just a few tenths of a percent is enough to cause a noticeable shift in frequency.

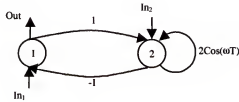


Figure 4-1 Single-weight oscillator network.

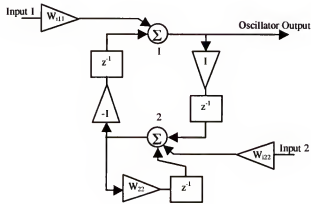


Figure 4-2 Single-weight oscillator architecture.

As a test case, a network was initialized to oscillate at 7 Hz. The sample rate was 216.99 samples/sec. The length of the sequence was 27 samples, a little less than a full cycle of oscillation. A target frequency of 7.2 Hz was selected, since only a subtle shift

in frequency was desired. Training was accomplished with standard BPTT. No momentum was used, although it is understood that the use of momentum terms during training would help maintain stability. The numerical sensitivity to  $W_{freq}$  also suggests that a very small learning rate be used during adaptation. The learning rate was set at a constant 0.001

A signal trace of the network outputs and desired response before training is shown in Figure 4-3. It required only three epochs to converge on the desired value for  $W_{freq}$ , with a frequency within 0.001 Hz of the desired response. A signal trace of the final network output and desired response after training is shown in Figure 4-4

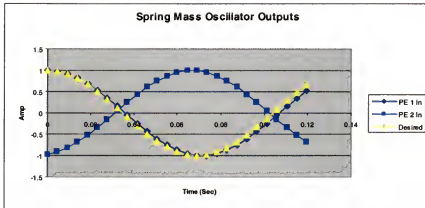


Figure 4-3 Initial signal trace of linear oscillator and desired response.

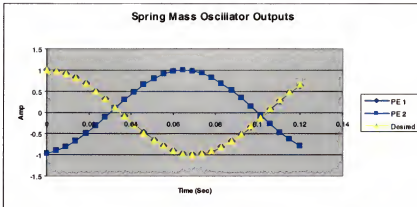


Figure 4-4 Trained signal trace of linear oscillator and desired response.

The small learning rate required to successfully train the network may be a limitation to the network's usefulness when used as a part of a larger, more complex network, requiring training over a larger solution space. Unless better training methods are found, there may be some tradeoffs made between accuracy and speed.

Fortunately there are alternatives that may help speed up training. Adjustable learning rates were tested, as were variable learning rates at different portions of the network. Good convergence was also accomplished when the weights were adjusted using an algorithm that moved the solution in the direction of the gradient by a small, fixed amount—Resilient backpropagation (RPROP).<sup>58</sup> In RPROP the size of the weight change varies depending on the slope of the learning curve, the sign of the gradient, and the frequency at which the gradient changes sign.

Resilient backpropagation was evaluated as a possible alternative, and was adapted to for use in combination with BPTT by using BPTT to determine the gradient, then updating the weight using the RPROP rules. It was determined that when the RPROP adaptation parameters were set to

$$\Delta_{\min} = 10^{-6}, \Delta_{\max} = 1.0, \eta_{\min} = 0.5, \eta_{\max} = 1.2, \Delta_0 = 10^{-3}, \quad (4-5)$$

adaptation was usually faster than when using BPTT, and in over 300 trials only two resulted in unstable weight values.

### Frequency Training

The attractive feature of this oscillator is that it uses a single weight to control frequency, so the actual frequency of oscillation can be directly determined from the final value of the adapted weight. In order to evaluate the uses of the network, many trials were run using signals that were in phase at the beginning of the sequence. Starting frequencies of 7 and 14 Hz were used. Signal lengths of one cycle, two cycles, four

cycles, and up to one second duration were evaluated. In phase and out-of-phase signals were looked at with the addition of a phase/gain compensating network, as will be discussed.

The practical limitations of finding the proper frequency weight value through adaptation are related to the nonlinear shape of the weight-to-frequency curve. It is interesting to visualize the function defining the weight. Figure 4-5 shows the entire curve as well as a closer look at the region for frequencies between five and twenty-five Hz.

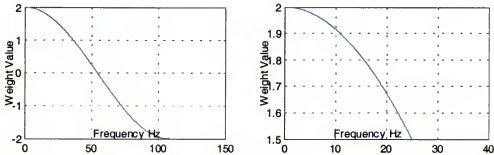


Figure 4-5 Oscillator frequency weight curves.

Notice that when adapting to frequencies at the lower end of the spectrum, the curve is flatter. The frequency is therefore more sensitive to very small changes in the value of the weight. As the desired frequency moves to the right on the curve, that sensitivity decreases. Adaptation trials showed that it usually took more epochs to find the proper weight value at the lower frequencies, as expected. This reinforces the notion that since weight values outside the bounds cause unstable results, the network must be initialized to oscillate in a stable region at the beginning of training, rather than the normal technique of weight randomization. It is best if a priori information can be used to determine an approximate frequency so that the search algorithm simply has to fine-tune the solution.

Oscillatory training signals pose another interesting problem. The MSE is normally used as the cost minimizing criteria during backpropagation. With sinusoids as desired responses, there are many conditions that can lead to a minimization of the cost function but not result in the proper solution. Solutions that lead to oscillation at harmonics of the desired frequency, for instance, will average the error to near zero. Also, signal lengths longer than a few cycles result in very small MSE, regardless of the frequencies involved. In fact, it was noted that as the length of the training signal increased, the ability of RPROP to correctly identify and shift the frequency was decreased. Table 4-1 shows some examples of the practical limitations of training the frequency-controlling weight to frequencies in the neighborhood of the starting value, as a function of the length of the training signal.

Table 4-1 Oscillator trainability within 1% desired frequency.

Training Signal Length	7 Hz Initial Frequency	14 Hz Initial Frequency
One Cycle	6 – 9 Hz	13 – 15 Hz
Two Cycles	6 – 9 Hz	12 – 18 Hz
One Second	6.3 – 7.7 Hz	13.3 – 14.7 Hz

Note that the spectrum is wider for the 14 Hz initial frequency tests when there are two cycles available for training rather than one. This is contrary to intuition—that longer sequences should be less successful, or more difficult to train. It turns out, though, that at the particular sample rate used (216.99 samples per second) there are very few samples to work with in a single cycle of oscillation at the higher frequency. It was, then, more difficult to converge on solutions away from the starting point. When more data became available—the two-cycle tests—the ability to train actually increased, implying that the optimum signal length for this frequency is larger than a single cycle. Determining the absolute optimal signal length to use was not the primary focus of this work. It is sufficient to assume that the proper value lies within the bounds tested.

Even though it was possible to successfully train the oscillator to follow a particular frequency from a given starting point, the input value of the second node was no longer optimal, and so the resulting signal was not an absolute exact match. Because of this, it was determined that perhaps if the criteria of the network were loosened to allow the scaling weight of the input of the second node,  $W_{i22}$  in Figure 4-6, to be trained as well a more exact signal match might be realized. This was evaluated and proved to be very successful. There were some unforeseen ramifications to this, however, that will be discussed later.

When  $W_{i22}$  is allowed to vary, the ability of the network to follow frequencies away from the initial point widens. It was observed in case after case that the frequency value and input weight went hand-in-hand when training to signals in phase with the starting frequency. Table 4-2 shows how the trainable frequency spectrum widens when  $W_{i22}$  is trained.

Table 4-2 Oscillator trainability within 1% desired frequency, plastic input weight.

Training Signal Length	7 Hz Initial Frequency	14 Hz Initial Frequency
One Cycle	3– 10z	8– 20z
Two Cycles	6 – 9 Hz	9– 18 Hz
One Second	6.3 – 7.7 Hz	13.3 – 14.7 Hz

### Phase and Gain Training

In order to evaluate the oscillator as an input to a larger system, a single-delay filter is now attached to the output of the oscillator in Figure 4-6.

Training now must be accomplished for the oscillator frequency weight, and the phase and gain weights,  $W_{i3,0}$  and  $W_{i3,1}$ , shown in the figure (as well as the input weight  $W_{i22}$ , if desired). But since the phase of the target signal can now be arbitrary, there is further complication in training the network. Signals with close frequencies, but offset

by a particular phase angle can also result in near zero MSE, while not converging on the proper solution. The trainability of the oscillator is therefore further reduces.

As an exercise in the visualization of the concept of trainability, the squared-differences between a beginning signal, say at 7 Hz, and other sinusoids at different frequencies and phases can be plotted. This is normalized by multiplying these differences by the expected value of the weights themselves, resulting in a surface that defines the error and whose slope represents the gradient that would be used in training. One can then assume that the regions of the surface with steeper slopes would be targets that would be easier to reach with gradient methods. Flatter portions of the surface represent areas in phase and frequency space that may be more difficult to reach, since the gradient provides less ‘kinetic’ energy towards the solution. Figure 4-7 illustrates this concept. This is not a depiction of the solution space, where a minimum can be visualized, but a diagram showing the magnitude of the gradient towards target solutions. It indicates very clearly that there may be frequencies or phases that may be difficult to reach during training.

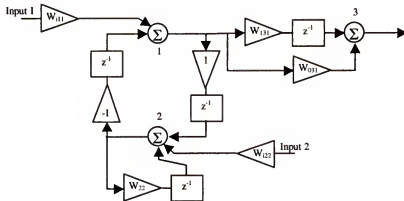


Figure 4-6 Complete single frequency oscillator architecture.

Figure 4-7 plots the difference between a 7 Hz sinusoid and sinusoids at frequencies between 3 and 15 Hz, at all phase angles between 0 and  $2\pi$ . It is scaled by the curve for the frequency weight (Figure 4.5). Notice that the plot indicates very little slope on the surface for all frequencies at relative phase angles of  $\pi/2$  and  $-\pi/2$ . This implies that the solution will be more difficult to resolve because the gradient is very shallow between the two sinusoids. On the other hand, phase angles around  $\pi/4$ ,  $3\pi/4$ , etc. should be easier to adapt to. Several test cases were run to measure the difficulty of adapting to specific relative phase angles, and the results were consistent with these expectations. Either finer training parameters—smaller step sizes or learning rates—or more epochs, or both, were required to reach a solution. And, as shown in Tables 4-1 and 4-2 above, some frequencies cannot be reached at all, in part because of this phenomenon.

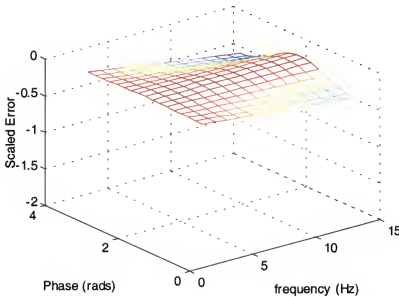


Figure 4-7 Gradient surface for 7 Hz initial frequency.



Another valuable training concept was realized by observing this graph. If it is known that the relative phase between signals may be between  $\pi/2$  and  $3\pi/2$ , it would be better to reverse the signs of the input vector, effectively shifting the initial phase by  $180^\circ$  making it easier to train to a viable solution. In later experiments this was done, and in one case was the only way to successfully track the desired response.

Similar analysis was done with a starting frequency of 14 Hz, whose weight value is located further down the curve (Figure 4-5) in a more linear region. Figure 4-8 shows its gradient surface. The differences between the 7 Hz and 14 Hz test cases are quite noticeable. As Table 4-2 shows, the trainable band for two cycles of oscillation for a 7 Hz initial frequency was about 4 Hz. For an initial frequency of 14 Hz it was 10 Hz.

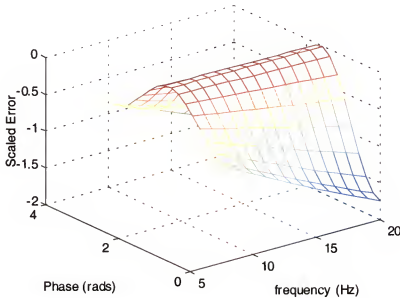


Figure 4-8 Gradient surface for 14 Hz initial frequency.

The number of epochs required to train the higher frequency oscillations was usually lower, although since not critical to the testing the number of epochs was not tracked with any detail. Notice, though, that the surface is still quite flat at relative

phases of  $\pi/2$  and  $3\pi/2$  similar to the 7 Hz initial case. So, the same limitations hold in the linear region: when signals are out of phase by these values they are more difficult to track.

### Sample Frequency Training Results

Two sample cases are presented to illustrate the ability of the oscillator to track a desired signal. The architecture of these examples is that shown in Figure 4-6, with the frequency weight  $W_{22}$  and the second input weight  $W_{i22}$  adapted. The initial oscillation and the desired response are in-phase at the start of the sequence. The first example case trains to a 9 Hz oscillation with the system initialized at 7 Hz, over about two cycles of oscillation. The second example case trains to a 14.5 Hz oscillation with the system initialized at 14 Hz, over a one-second signal length. The results presented represent typical results of about 300 tests run in support of this analysis.

The results for the first example are shown in Figure 4-9. The figure contains a signal tracing plot, showing the desired response overlaid with the final oscillator output, a plot showing the PSD of both signals, and the learning curve, MSE as a function of epochs. The PSD is provided for comparison only, since the bin size of the FFT may not be large enough to indicate the correct exact frequency.

This first case is typical of training the oscillator to signals in-phase with frequencies in the neighborhood of the initial value. The network was initialized to 7 Hz and trained using RPROP with gradients backpropagated through time. The training curve shows that a reasonable solution was found in very few epochs, and that the absolute minimum training criteria of  $10^{-8} \geq \text{MSE}$  was met after about 100 epochs.

The second case is typical of results of signal lengths of one second. Figure 4-10 shows the signal trace, PSD and learning curve. The signal traces are indistinguishable.

The learning curve shows that a viable solution was obtained after just eight epochs, and the absolute minimum was found in only 45 epochs. These two cases were chosen to illustrate the level of accuracy that can be obtained under the right conditions for training the oscillation.

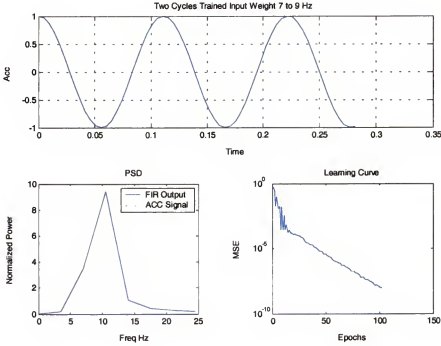


Figure 4-9 Freq. training test, two cycles. Trained input weight, 7 to 9 Hz.

### Dual Sinusoid Phase and Gain Training Example

The primary purpose of this study, of course, is to provide an oscillatory input to a larger network intended to replicate the behavior of system with energy at many frequencies. To evaluate the ability of the oscillator and its phase/gain filter to do this, tests were accomplished training a larger network to track a signal with two sinusoids at different relative phase angles. Equation 4-6 describes the desired response.

$$1.2\cos(2\pi 7.5t + \phi_{7.5}) + 0.7\cos(2\pi 14.5t + \phi_{14.5}) \quad (4-6)$$

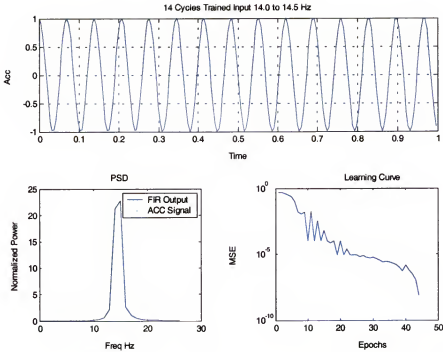


Figure 4-10 Freq. training test, one-second. Trained input weight, 14.0 to 14.5 Hz.

The variations on phase angles  $\phi_{7.5}$  and  $\phi_{14.5}$  are shown in Table 4-3. These phase values were chosen to challenge the network by looking at relative angles all around the phase circle, and specifically to look at angles that were expected to be difficult to follow, based on the previous analysis. Testing was done for signal lengths of two and four cycles of the lower frequency, and one second total time.

The network used is shown in Figure 4-11. It consists of two linear oscillators, each with a phase/gain filter attached to its output. The filtered signals are then summed. Oscillator number one, indicated by nodes 1 and 2 in the figure, was initialize to 7 Hz, the other was initialized to 14 Hz. Both had unity gain and started in phase. The frequency weights  $W_{22}$  and  $W_{44}$  were trained, as were the phase/gain weights going into node 5 for both oscillators. The input weights  $W_{i22}$  and  $W_{i44}$  were trained as well.

Table 4-3 Dual sinusoidal frequency phase/gain test phase values.

Test Number	$\phi_{7.5}$	$\phi_{14.5}$	Input weight sign change
1	$\pi/8$	$-\pi/4$	No change
2	$3\pi/8$	$-\pi/2$	No change
3	$5\pi/8$	$-3\pi/4$	No change
4	$7\pi/8$	$-7\pi/8$	No change
5	$5\pi/8$	$-3\pi/4$	Input weight signs negated
6	$7\pi/8$	$-7\pi/8$	Input weight signs negated

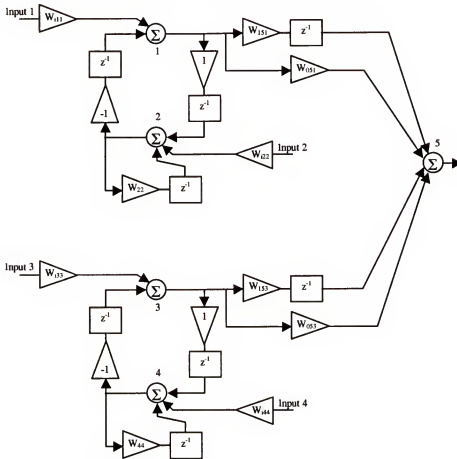


Figure 4-11 Complete double-frequency oscillator architecture.

### Dual Sinusoid Phase and Gain Training Example Results

Test number 5 was chosen to be an example in this analysis since it exhibited the most difficult training problems encountered during the experiments. The phase angle of the lower frequency is just past the  $\pi/2$  hump in the gradient curve, as is the angle of the

higher frequency, but on an opposite slope. The target signal was long enough to contain four cycles of the original 7 Hz initialized frequency of the first oscillator.

Backpropagation through time modified RPROP, as previously described, was exclusively used for training. Initial input values were negatives of those originally defined, in order to start the oscillators on the correct side of the phase circle. This was necessary because training was not successful with positive inputs, as was suspected would be the case from the previous analysis.

Figure 4-12 shows the final results for Test 5. Notice that the desired signal and the network output are indistinguishable in the signal trace, as well as the PSD. Note the learning curve. A very reasonable solution was found after the first few epochs, and learning continued towards the limits at the end of 1000 epochs—the maximum allowed for training in these tests. Perhaps more interesting to note is the information in Figure 4-13, however.

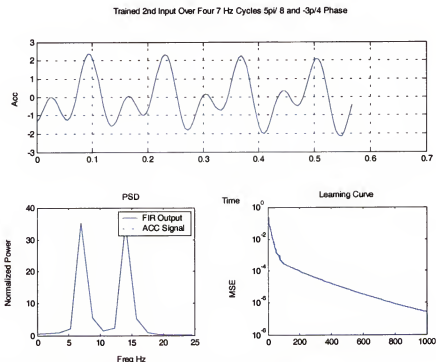


Figure 4-12 Dual frequency output with trained input weights.

Epochs

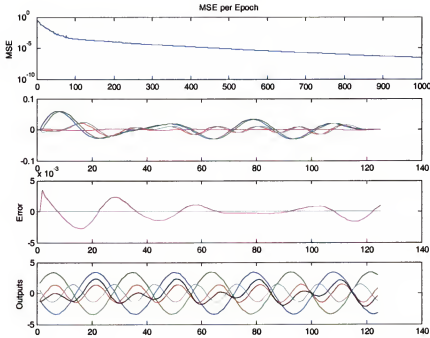


Figure 4-13 Dual frequency with trained input weights training results.

Figure 4-13 is a capture of a combination of traces that are output during training. The capture shown is that of the final epochs of training. The first trace is an expanded version of the learning curve as training progresses. The second shows the gradients for each node output. The third is a plot of raw signal-to-target error at the network output. The final plot is a trace of the outputs of each node.

Most interesting to note is the traces of the outputs from each individual node. It is difficult to see in the plot, but the trace of the output can be seen, as can the positive and negative oscillations at each of the two frequencies. Most telling is the fact that the maximum amplitude of the oscillators is no longer contained between  $\pm 1$ . Allowing the second input to be scaled has resulted in a relaxation of the true unitary oscillatory nature of the network. While it's true that in theory the phase/gain filter has all the capability required to shift the phase and gain by itself, it appears that the training process has used

the added degree of freedom in the plasticity of the input weight to help shift the phase of each oscillator. The final output is still entirely correct, but the network can no longer be evaluated as separate pieces of a system.

This discovery lead to another experiment. Suppose all the input weights are fixed, and only the frequency and FIR filter weights allowed to be adjusted each epoch. By doing so the optimal value of the frequency weight can be identified. Once the frequency weight is found, the correct value of the input weight can be calculated and fixed. The process can be repeated epoch by epoch until optimum performance is realized.

This was accomplished for the six test cases in Table 4-3. Again, Test 5 proved most interesting. Figure 4-14 shows the results of this test. Notice that the signal tracings are no longer exactly matched, and that the learning curve levels out at a MSE value much higher than before. Also, the energy levels in the PSD are subtly different.

The behavior of the network components is more evident in Figure 4-15. This composite graph was again captured at the last epoch of training. The gradient values are substantially higher, but training essentially stopped. This indicates that the capability of the network to completely compensate for phase and gain is less adequate. This is understandable since the degrees of freedom available have been reduced.

Understandably, then, the raw network error is also higher. But the outputs from each node in the network, traced in the fourth graph, show that each oscillator is now behaving exactly as expected, within  $\pm 1$ .



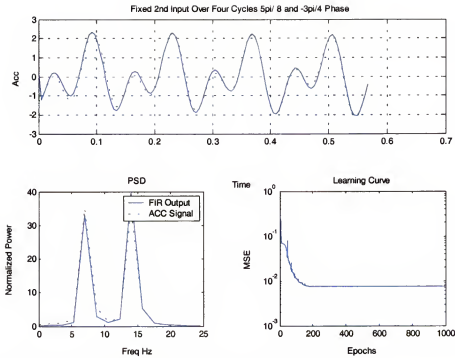


Figure 4-14 Dual frequency output with fixed input weights.

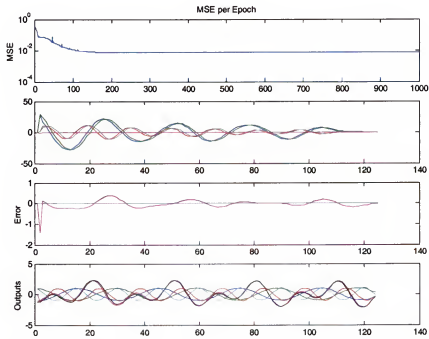


Figure 4-15 Dual Frequency output with no trained input training results.

A process has now been developed by which the frequency, phase and gain can all be adjusted, while maintaining the purity of the oscillators themselves.

### Post Analysis Oscillator Design Modifications

The oscillatory circuit was been tested at various signal lengths, with occasional instabilities in some of the shorter signal duration times were noted. Analysis showed that some of the frequency weight values had drifted past the 2.0 maximum for stability. This drove two design changes to the network—one directly and one indirectly.

The direct design change was to the linear nodes used in the oscillator circuit. The summing nodes are simply linear nodes. These were changed to be symmetrically limited linear nodes whose output function is shown in Figure 4-16, and is mathematically described as

$$f(x) = \begin{cases} 1.1 & x > 1.1 \\ x & -1.1 < x < 1.1 \\ -1.1 & x < -1.1 \end{cases} \quad (4-7)$$

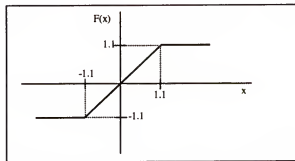


Figure 4-16 Modified saturated linear activation function.

This ensures that the output of the oscillator is always limited, but since that limitation is within  $\pm 1.1$ , the network could no longer be initialize such that the oscillator outputs were scaled by the mode amplitudes. Initialization and modification of the modal amplitude will now have to be entirely adjusted by the linear filter output stage of the

network. It should be noted that the values  $\pm 1.1$  rather than  $\pm 1.0$  for the limits of the node are used because the output of the oscillator ‘bleeds’ beyond  $\pm 1.0$  from time to time because of computational approximations. Clipping of the oscillator output was noted, a phenomenon that degraded the ability of the network to train well.

### **Linear Oscillator Summary**

Under the correct conditions the oscillator is tunable to frequencies in the region of interest to the overall project. It was shown that the network needs to be initialized correctly for successful training. The frequency weight must be set in the neighborhood of the expected final value. Guidelines for the expected bandwidth for successful training showed that if the value of the desired frequency is known within a few percentage points a long sequence—up to one second—can be used as a target signal. If wider frequency resolution is required shorter sequences will have to be used—one or two cycles of oscillation.

The correct phase and gain can be found for signals whose values are arbitrary, given that enough degrees of freedom are allowed in the network, and that an initial ‘good guess’ is available for the frequency weight. It was shown that there is a tradeoff between overall accuracy and pure oscillatory behavior in the network when the input weight to the second node is allowed to vary.

A process by which the maximum flexibility in the larger phase/gain network can be used to train the frequency variable was outlined. The user must make the decision whether or not to then adjust the oscillator input weights at the end of each epoch’s training as a function of the adjusted frequency weights, or train them directly. For this work the more desirable training method is to hold the inputs fixed while adjusting the

frequency weight, then update the input weights to match. This maintains the integrity of the oscillators as pure sinusoidal signal generators.

### **Simplified Generalized Likelihood Ratio Test for Detecting Signal Changes**

The second part of this chapter looks at a series of experiments were designed and conducted to evaluate the practical benefits and limitations of the simplified GLRT for detecting changes in an LCO desired response. The intent was to test the algorithmic concepts and build a practical paradigm based on lessons learned in each preceding experiment. The goal was to design trials that would build upon results of each preceding example, culminating in a fully defined and functioning algorithm for signal change detection, switching and training. The experimental signals used throughout were artificial bimodal LCO responses. All experiments center around the network topology used in the bimodal test in the previous section as depicted in Figure 4-11, including a single-delay FIR output filter for each mode. The oscillators are initialized to provide sine waves by setting the input weights

$$W_{i1} = 0; \quad W_{i2} = -\sin(2\pi f_i T_s). \quad (4-8)$$

where  $f_i$  is the frequency of oscillation in Hz, and  $T_s$  the sample time. The frequencies  $f_i$  were set to match those of the desired response. The simplified GLRT was used to provide the detection statistic to direct model switching for segmenting and modeling the desired response.

Seven experiments were conducted, beginning with a simple test of the algorithm itself, becoming progressively complex with each successive experiment. A basic two-frequency sinusoidal signal whose frequencies were fixed was used as a desired response. The sinusoids were in phase with respect to each other in earlier experiments but were shifted to enhance realism in later tests. The relative amplitudes of both frequencies

remained constant, but the overall amplitude was a function of an independent variable  $m$ , described by a discrete set of values and contained in an  $m$ -vector. The desired response  $d_s(t)$  for segment number  $s = 1, 2, \dots$ , corresponding to the position of  $m$  in the  $m$ -vector, is described by

$$d_s(t) = A_1(Km + c)\sin(2\pi 7t + \phi_1) + A_2(Km + c)\sin(2\pi 14t + \phi_2). \quad (4-9)$$

The constants were set as:  $A_1 = 0.5$ ;  $A_2 = 0.1666$ ;  $K = 20/3$ ; and  $c = 0.1$ . The vector  $t$  is a sequence of discrete times,  $t = nT_s$ ,  $n = 0, 1, 2, \dots, N-1$ , where  $N$  is the total number of samples and  $T_s = 1/216.99$ , the sample time. The phase vector  $\phi = [\phi_1 \ \phi_2]$  was initially set to  $\phi = [0 \ 0]$  until later experiments. The  $m$ -values were either switched by a random selection of  $s$ -values, or arranged so as to provide a desired response with a particular behavior. When  $s$ -values were randomly selected, the time between changes was held within a certain minimum,  $t_{\min}$ , to allow statistical stability in the simplified GLRT.

Neural network models were made available to the system with parameters set to match segments of the desired response. As a minimum, each experiment had one model corresponding to the first segment. As a general rule during the experiments, if no existing model was sufficient, the closest model to the desired response, based on minimum MSE, was used as an initial network parametric set for training a new model.

Networks were trained using BPTT simplified RPROP with parameters set as:  $\Delta_{\min} = 10^{-9}$ ;  $\Delta_{\max} = 1.0$ ;  $\eta_{\min} = 0.5$ ;  $\eta_{\max} = 1.2$ ; and  $\Delta_0 = 10^{-6}$ . Training stopped when the MSE of the desired response to the network output decreased to a sufficiently small value, usually  $10^{-3}$ , or until the learning curve showed little progress as determined by averaging the change in gradients over a window of ten epochs from the current epoch

back. The resulting slope was used to project the epoch number at which the learning curve could be expected to decrease below the stopping criteria. If the projected final epoch number remained greater than the maximum allowable for ten consecutive epochs the learning curve was deemed flat and training stopped.

Since each experiment is intended to provide further insight into the construction of a practical data synthesis algorithm, the results from previous experiments help define subsequent experiments. Therefore, in order to maintain clarity each experiment and its results will be described together in sequence.

### **Preliminary Experiments**

An initial set of three experiments were designed and conducted to establish the viability and limitations of the simplified GLRT and the proposed network topology as part of an algorithm for multi-modal LCO data synthesis. Since these are simple tests to establish the foundations for simplified GLRT use they won't be discussed in detail. The information gained from them, however, will briefly be presented. All three used the same set of eight variations of bimodal signals as described by Eq. 4-9.

The first experiment tested the initial use of simplified GLRT to detect changes in an oscillatory system. Models for all variations of the desired response were available. The minimum time between random changes in a response was set at one second to allow plenty of time for the statistics to stabilize. Results showed that simplified GLRT was a viable detection statistical method, detecting changes in the response usually within one data frame.

The second preliminary experiment was intended to evaluate the use of simplified GLRT when changes occurred at relatively rapid rates. Trial and error throughout the initial stages of evaluating actual flight test data established that networks need a

minimum of 0.25 seconds of data, about 54 samples, to be successfully trained. This experiment was set up to determine if changes at that rate could be detected. Models for all possible variations of the desired response were again available.

Three valuable pieces of information were gained from this experiment. First, it was shown that simplified GLRT can detect rapid signal changes, at least when good models exist for all possible permutations. Second, an algorithm for dealing with the practical aspects of L-values was developed. As L-value plots will indicate, the quality and behavior of L-values is related to the ability of the model being examined to match a desired response. Models closely matching a response result in steeply increasing L-values that rise to a distinct peak at the point of model change. As the model fidelity decreases the L-values become lower and less defined. Subsequently decisions based on L-values become more difficult. The threshold used to make the decisions can no longer be fixed at a high value. As a result of this preliminary work an algorithm was developed whereby the threshold was preset at an arbitrarily high number, say  $\text{Threshold} = 400$ . If no L values break the threshold the value is decrease by 20% until it reaches a minimum value based on the number of data samples or the threshold is broken. A lower bound for the L-value decision threshold was set to be  $2.3N$  determined through trial and error.

The third lesson learned from this experiment deals with the practical minimum amount of data required for signal change detection and modeling. It is now known that the network oscillators train best when the phases of the desired response and those of the oscillators are in the first quadrant. Since the oscillators are set up as sine waves, it was anticipated that it might be necessary to begin a data window at a point just after a zero-crossing on a positive slope, thus reducing the data set in each window by a few sample points in the beginning. To compensate, an additional number of samples must be added

to each data window approximately equal to the number of samples in a single oscillation of the lowest frequency contained in the desired response; 7 Hz in these experiments.

Therefore 31 extra data points were added, making the value for  $t_{\min} \approx 0.40$  seconds. This was the value used in all subsequent experiments.

The final preliminary experiment was intended to test the algorithm's ability to use simplified GLRT to detect changes, identify the corresponding change time, and to identify the need for training or fine-tuning a model when no models in the database are sufficient. The desired response was made up of the same eight variations of a sum of sine waves as in prior tests, but the network database contained only four of them. Changes in amplitude were made at random after a minimum amount of time had expired since a previous change,  $t_{\min} \approx 0.40$  seconds. This meant that in order to be successful the algorithm must identify both the time of a change and the closest model to the desired response after that change, and if insufficient, train a new model and add the results to the model database. The experiment was a success. The algorithm using the simplified GLRT was able to properly detect a signal change and the correct model was identified by the lowest MSE, or a new model was successfully trained, stored in the model database and used later as required.

This preliminary work established the working parameters and guidelines for building a practical system. The following four experiments take these concepts and progressively build upon them to establish a complete multi-modal LCO data synthesis system.

### **Experiment 1**

Experiment 1 was designed to test the algorithm's ability to detect small, incremental changes in a desired response. The test signal was constructed as a gradually



increasing response with  $m$ -values incrementally increasing from  $m = 0.6$  to  $m = 0.975$  in step sizes  $\Delta m = 0.025$  as shown in Figure 4-17. Once the signal reaches the highest allowable  $m$ -value it is decreased slightly and then increased again, simulating the response of an LCO system whose parameters are changing slightly over time.

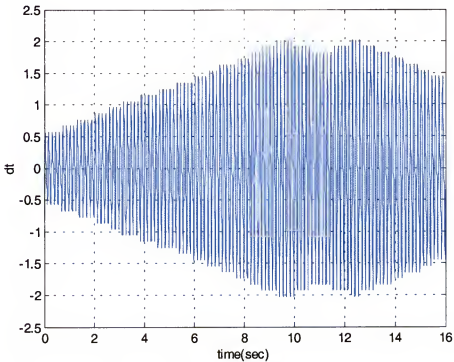


Figure 4-17 Experiment 1 test signal.

The  $s$ -vector,  $s = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15\ 16\ 15\ 14\ 15\ 16\ 15\ 14\ 13\ 12\ 11\ 10]$  is not chosen at random in this experiment. To further challenge the system, only models corresponding to four of the sixteen amplitude levels at  $m' = [0.6\ 0.7\ 0.8\ 0.9]$  are contained in the initial model database. The original signal levels and the corresponding model database locations are given in Table 4-4. The model number corresponding to each desired response segment and the model number detected by minimum MSE are shown in the rows labeled 'Model Number,' and 'Detected Model Number' respectively.

The actual time of the change is in rows labeled 'Time,' and the detected switch time in the rows labeled 'Detected Time.' The rows labeled 'Error (frames)' give the difference in actual time and detected time in sample time-steps. Blank table cells indicate that no change was detected. The original four states contained in the database, A, B, C and D are highlighted in boldface type for clarity.

Table 4-4 Experiment 1 desired and ANN models.

m-value	Signal level	ANN model (bold = original)
0.6	1	<b>A</b>
0.625	2	E
0.65	3	F
0.675	4	G
0.7	5	<b>B</b>
0.725	6	H
0.75	7	I
0.775	8	J
0.8	9	<b>C</b>
0.825	10	K
0.85	11	L
0.875	12	M
0.9	13	<b>D</b>
0.925	14	N
0.95	15	O
0.975	16	P

Results of Experiment 1 are provided in Table 4-5 and L-values illustrated in Figure 4-18. From the L-value plot it is seen that all known models were easily identified. This is also evident in Table 4-5. The algorithm successfully trained and stored neural models as required, incrementally as expected. All changes times were identified within a few frames of their actual occurrence, most within one frame. There are two interesting exceptions. The transition between segments 15 and 16 ( $t = 9.277$  sec) was detected 3 frames late. This is acceptable, but the cause is interesting. The L-values at that transition are very low, and the trailing edge of the curve is not a sharp decrease but a more gradual slope. This made it more difficult for the algorithm to detect

a true peak since two sharp changes in the L-values are actually present. The second point of interest occurs between the 20<sup>th</sup> and 21<sup>st</sup> segments ( $t = 13.226$  sec). This transition was detected earlier than it actually occurred. Again, close inspection of the L-values of these two segments show that the curve is not sharp but more gradual, making an exact determination of the transition time difficult to determine. Since the error amounts to little more than 16 ms in actual time it is an acceptable error.

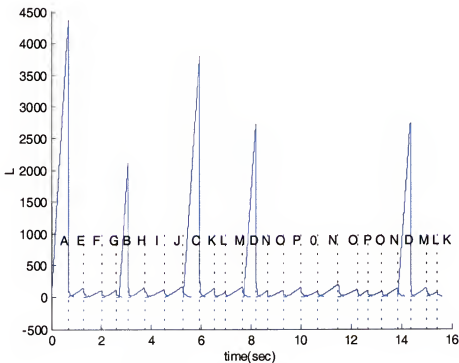


Figure 4-18 Experiment 1 L-values.

The algorithm was successful in training new models as necessary between known models. Those new models were of sufficient fidelity as to ensure they could be reused as needed. This is illustrated in the portion of the desired response that decreases slightly from its maximum between  $t = 9$  and  $t = 13$  seconds. Models trained and stored as the signal had increased prior to the peak were successfully reused. The algorithm did not

show any 'false alarms' causing unnecessary retraining. This is a significant result because it indicates that the algorithm is capable of maintaining its efficiency.

Table 4-5 Experiment 1 Results.

Model Number	1	2	3	4	5	6	7	8
Detected Model Number	A	E	F	G	B	H	I	J
Time	6.775	1.286	1.995	2.590	3.065	3.747	4.539	5.263
Detected Time	0.682	1.290	2.005	2.595	3.069	3.751	4.544	5.268
Error (frames)	1	1	2	1	1	1	1	1
Model Number	9	10	11	12	13	14	15	16
Detected Model Number	C	K	L	M	D	N	O	P
Time	5.940	6.535	6.982	7.650	8.194	8.650	9.277	10.005
Detected Time	5.945	6.539	6.986	7.655	8.199	8.655	9.291	10.009
Error (frames)	1	1	1	1	1	1	3	1
Model Number	15	14	15	16	15	14	13	12
Detected Model Number	O	N	O	P	O	N	D	M
Time	10.678	11.475	12.259	12.683	13.226	13.849	14.351	14.991
Detected Time	10.683	11.480	12.263	12.687	13.208	13.853	14.355	14.996
Error (frames)	1	1	1	1	-4	1	1	1
Model Number	11	10						
Detected Model Number	L	K						
Time	15.411	16.000						
Detected Time	15.415	End of data						
Error (frames)	1							

Experiment 1 was very successful in demonstrating the algorithm's ability to detect small, incremental changes in a desired response, and train and store new neural models as necessary to successfully replicate the signal. It was also shown that new neural models once stored could be successfully utilized again without the need for retraining.

### **Experiment 2**

Experiment 2 begins to approach a realization of a system that can be used against a continually changing response. The proposed LLM system is a linear, stepwise approximation to a continuous response. It is therefore necessary to test the algorithm on such a response in order to determine its ability to detect when signal levels have changed sufficiently to require a different model.

The desired response for this experiment was a bi-modal sinusoid with both modes in phase using Eq. 4-9. The time-span for the response was  $0 \leq t \leq 16$  sec., in increments of the sample time  $T_s$ . The  $m$ -values used increased linearly as  $t$ ,  $0.6 \leq m \leq 0.975$  in increments of  $\Delta m \approx 0.000108$ . Neural models, 16 in all, corresponding to the desired response at  $m = [0.6 \ 0.625 \ 0.65 \ 0.675 \ 0.7 \ 0.725 \ 0.75 \ 0.775 \ 0.8 \ 0.825 \ 0.85 \ 0.875 \ 0.9 \ 0.925 \ 0.950 \ 0.975]$  are available to the system. It was hoped that these increments in  $m$  would be fine enough to prevent retraining.

As expected the system needed to continuously decrease the  $L$ -value threshold in order to detect a desired response change sufficient to warrant a model change. Results of Experiment 2 are shown in Figure 4-19. The desired response is overlaid with the output from each selected neural model. As can be seen, the predicted response is a stair-step approximation to the actual response. The  $L$ -values, shown in Figure 4-20, no longer behave as a set of clearly defined ramps, but instead are bell-shaped curves. The model change time is indicated at the point at the top of each 'hill' rather than by a steep

decline in values as before. The L-values are now very low. This is expected since there is no one model that matches the response at any given time. Instead the simplified GLRT is measuring how quickly the current model's ability to match the desired response is decreasing. When the current model can no longer keep up, in other words when statistical difference between the null and posited hypotheses becomes insignificant, the algorithm begins searching for a matching model.

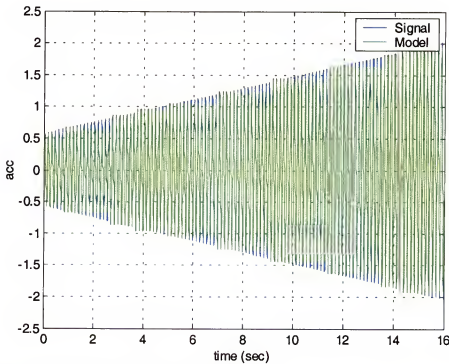


Figure 4-19 Experiment 2 test signal comparison.

An interesting phenomenon can be seen in Figure 4-20. Notice that a change in model, or perhaps more precisely the signal-change hypothesis was detected between the second and third segments but the model number didn't change. This behavior is also noted four other times during the experiment. The maximum allowable MSE value for this experiment was  $10^{-3}$ . The behavior is interesting because it illustrates that even though the L-values showed a possible model-change point, the criteria set for an

acceptable model was such that the current model was still sufficient. In this experiment having models available at  $m$  increments of 0.025 was overly precise. Larger increments in the known models would have been sufficient. Alternately, of course, these results indicate that the criteria for an acceptable model could have been lowered.

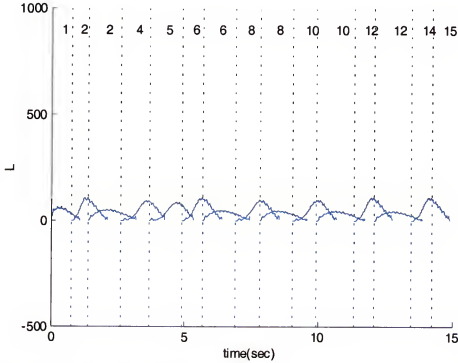


Figure 4-20 Experiment 2 L-values.

### Experiment 3

Experiment 3 uses the same desired response and neural model setup as Experiment 2 except that the relative phases of the two modes in the desired response are now shifted. The vector phase vector in Eq. 4-9 is now  $\phi = \begin{bmatrix} \frac{\pi}{4} & \frac{\pi}{2} \end{bmatrix}$ .

Figure 4-21 shows the desired response overlaid with the system output. Again the predicted response is a stair-step approximation following the desired response very closely. The maximum L-values are shown in Figure 4-22. Again notice that there are

instances where the algorithm detects a possible model change, but the MSE criteria allows the current model to stay in place. The only difference between the results of Experiments 2 and 3 are the locations of these model overlaps, due to the phase changes in the desired response.

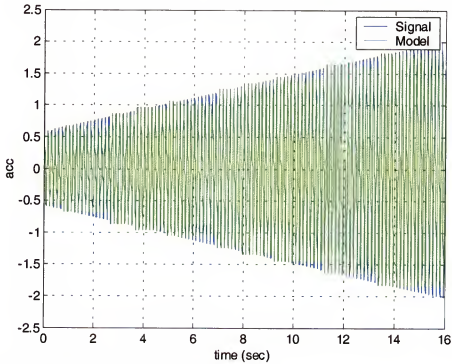


Figure 4-21 Experiment 3 test signal comparison.

These results give confidence in the algorithm's ability to segment and model a continuously changing system, identify the possible need for a change in model for signal approximation, and identify which model in the database provides the best approximation. The final test of the algorithm, then, is one requiring training and storage of new models.



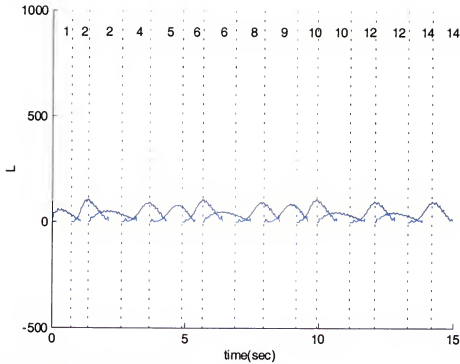


Figure 4-22 Experiment 3 L-values.

#### Experiment 4

The simplified GLRT evaluation culminates in an experiment that tests the proposed algorithm's ability not only to detect changes in a continuously changing signal, but also to successfully train and store neural models as necessary to continue approximation of the signal by the system. The desired response for Experiment 4 is the same as used in experiment 3. A neural model for the desired response at time  $t = 0$  is the only model contained in the database, however. It is up to the algorithm to detect a signal change significant to warrant a change in model, and to train that model.

Training presented an interesting challenge. The topology of the network is such that its output is a modal LCO response with constant amplitude. But the desired response does not have fixed amplitude, so any model trained to follow the response will be an approximation regardless of the size of data window used to select the training data.

Since the network cannot directly account for continuous amplitude changes, it tends to compensate by shifting one or more of the degrees of freedom available to it. This tendency is reflected in a slight increase in frequency or phase, depending on which network parameters are adjustable and which are fixed. Consequently it is desirable to use a small amount of data for network training to minimize these errors. However, fidelity is lost as the amount of data available for training is decreased, so training becomes a constant tradeoff of parametric accuracy versus overall MSE minimization.

Results of Experiment 4 are shown in Figure 4-23. Figure 4-24 shows the L-values. Notice that the desired response and the output of the system match very closely throughout the entire experiment. Still visible is the stair-step approximation, but the overall fit is extremely close at all points. Also interesting is the L-value plot. Notice that as before, model changes were indicated, and in this experiment new models trained and stored, but there were several times when switching to a new model was not necessary due to MSE criteria. All but the first model was used twice by the system before it became necessary to retrain. This also illustrates the system's ability to store models and recognize their utility again later in the response.

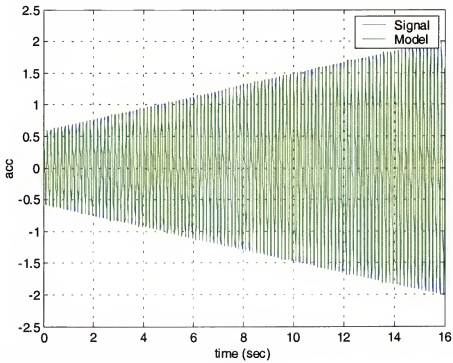


Figure 4-23 Experiment 4 test signal comparison.

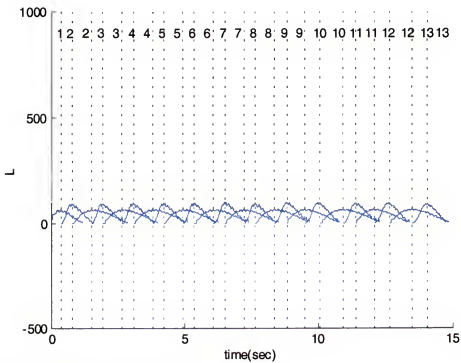


Figure 4-24 Experiment 4 L-values.

The total desired response signal duration was 16 seconds. A total of 13 models were necessary to approximate that signal with the neural system, indicating that on the average a model was accurate for more than one second. As can be seen in Figure 4-23, the one-second approximation is consistent across the entire desired response.

Experiment 4 successfully demonstrated that the use of a simplified GLRT as a part of an LCO synthesizing system initialized with a single model and could be used to monitor and train models for close approximation of a continuously changing LCO response. Further, it showed that LLMs are sufficient for modeling at least one second of modal data.

### **Network Parametric Evaluation**

Experiments 3 and 4 used identical desired responses. The only difference was that experiment 3 switched between known models, while experiment 4 trained models to follow a given portion of signal. A comparison of the network parameters to the actual constantly changing network parameters used in each method as a function of time is insightful.

It was assumed that a method that used stored models and selected the closest to the response would operate faster than a method that trained new models as required. The assumption proved to be accurate. The training method took on the order of 100 times longer to run over the entire 16-second signal duration than model switching. If time is not necessarily the most important factor, the question becomes one of accuracy. Intuitively it would seem that training models to fit specific portions of a response would be more accurate than using predetermined models.

Figure 4-25 is a plot showing the FIR parameters for each mode in the actual continuous response and the fixed-parameter switched networks in Experiment 3. The

parameters are normalized so that the absolute values of the FIR vectors are illustrated. Visible are the stair-step approximations to parameters as they proceed up the ramp representing the actual FIR magnitude for each mode. It can be seen that the approximations tend to cross back and forth over the line, indicating the give-and-take in MSE as models change. The weights tend to be selected higher than the actual solution, then drift back under the line. This is a phenomenon caused by the linearly increasing amplitude of the desired response. Consequently the nearest of the available models corresponded to a model whose amplitude was higher than that of the desired response at the switch time.

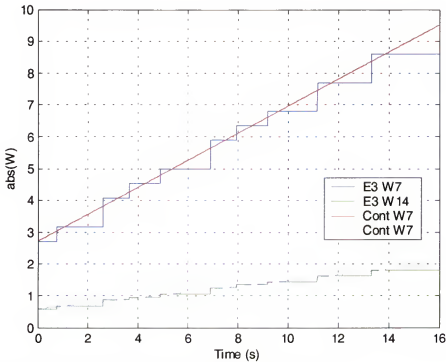


Figure 4-25 Experiment 3 network parameter comparison.

A similar analysis of the network parameters found in Experiment 4 is provided in Figure 4-26. Again the stair-step approximation is clearly visible, but in this case there is a bias in the curve towards the inside of the desired response, indicative of the trained

response's tendency to follow the earlier portion of data used in training slightly better than data later in the training sequence because of the increasing amplitude and the inherent weighting of training data in the backpropagation through time process. The result is that the training method matches the amplitude of the desired response closely at the training time. This is the opposite of the tendency noted in the model switching method.

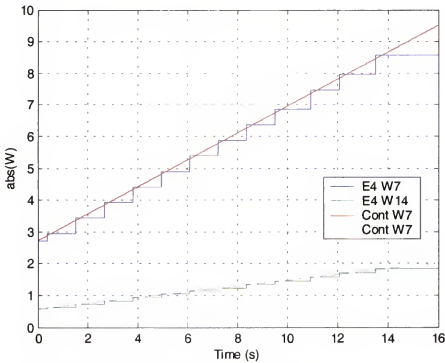


Figure 4-26 Experiment 4 network parameter comparison.

The most important comparison between the two methods is the number of model changes necessary to replicate the desired response. The model switching method was able to accomplish the task with nine model changes. The training method required twelve. This gives the model switching method a significant advantage since model training times can be quite long. Therefore if a sufficient model database exists, model switching is clearly the preferred method.

### Reduced-Model Simplified GLRT Experiment Conclusions

Results of Experiments 3 and 4 have demonstrated a method by which a multi-modal, varying LCO response can be modeled using properly configured and selected simple adaptive systems. The goal was to determine if a series of artificial neural networks could be constructed and trained to synthesize an LCO response while providing further insight into the nature of the LCO behavior by analysis of network parameters and their changes over the duration of a desired response. An algorithm that uses a detection statistic to measure changes in a desired response and direct the use of simple neural networks whose parameters can be selected from a set or retrained as necessary to track segments of the LCO responses accomplished this.

These experiments were intended to aid in the step-by-step design of this algorithm. The results of each experiment were applied to the next so that lessons learned during the process were immediately incorporated into the paradigm. This proved to be a very viable approach, resulting in an algorithm that was successful in all aspects.

It can be concluded that the use of the simplified GLRT as a detection statistic works well for this task. It was shown that the simplified GLRT could detect rapid and subtle changes in a response within a sufficient amount of time, usually within one data frame of the actual signal change, to allow an algorithm to properly select or train the parameters of a simple neural network to match the desired response over a finite period of time. It was shown that for an experimental multi-modal response, detection and training could be accomplished with a training sequence of 54 samples providing a mean square error value less than or equal to  $10^{-3}$ .

A comparison of two methods for utilizing the networks was made. The first method, network switching, stores predetermined network parameters and selects the

proper set for a given segment of a desired response. The second, network adapting, begins the process with a predetermined set of network parameters but adapts new parameters as necessary. Both methods are directed by the simplified GLRT. It was shown that both methods provided system outputs that matched the entire desired response very well.

The determining factors in a comparison of the two methods are execution time and efficiency. The network switching method ran on the order of 100 times faster than the network training method. It required nine model changes to track the desired response as opposed to twelve changes required by the training method. It must be concluded, therefore, that for a practical system the method of network switching is the optimal method to use given that its output can maintain the criteria established for acceptable error.

It is certainly understood that populating a model database with sufficient models prior to an analysis is a difficult challenge, and in all likelihood impractical. But it is possible to form some preliminary approximations that can be used as a starting point. Therefore, it is concluded that a practical method for data synthesis of a multi-modal, varying LCO response uses the simplified GLRT as a detection statistic directing the use of a network switching algorithm with a database of stored network parameters, allowing training and storage of new parameters as necessary. The modular LLMs run continuously, providing time-series responses for all models in the database for fast comparison to the desired response. Model parameters can be analyzed individually or as sets providing direct measurement of modal frequency, phase and gain as functions of time and any independent variables that may be influencing a desired response.



### **Fixed-Length Flight Test Signal Training Experiments**

The final section of this chapter begins the process of using actual flight test data to complete the data synthesis paradigm. Fixed length signals are used to measure and define the practical limits for the various parameters used.

#### **Preliminary Work: Constant Mach, Altitude and Normal Acceleration LCO Evaluation (Test 8c)**

In order to complete a paradigm for simulating LCO data, consideration must be given to the practical limits of the LLMs. Since the goal is to match flight test data it makes sense to use portions of that data to help more fully define the models, as well as to establish the training parameters for the LLM as a complete set. To begin these evaluations an interesting portion of data from the Test line 3 data set was used as a basis for some preliminary experiments. Data from the left wing forward wing tip accelerometer taken during Test 8 serves as the desired response. Specifically the data is labeled Test 8c, as described in Chapter 2. This is a twelve-second portion of data showing nearly constant amplitude LCO after the effects of an artificial oscillatory input from the flutter excitation system have begun to damp. The data from Test 8c is suitable because it contains well behaved LCO while at the same time displaying some nonlinear characteristics such as subtle changes in amplitude and frequency. Its length is long enough to allow evaluation of extended desired responses, but short enough for practical training.

As a starting point for a model, the oscillator driven single-delay FIR architecture used to evaluate bi-modal signals in the simplified GLRT experiments is expanded to include 16 oscillator pairs, as shown in Figure 4-27, providing modal outputs for the 16 modes identified in linear flutter analysis. The oscillators are initialized to the free

vibration frequencies (Table 4-6) since those frequencies are assumed to be in close proximity to the actual frequencies contained in the signal.

Table 4-6 Initial Modal Frequencies (free vibration values).

Mode	1	2	3	4	5	6	7	8
Freq	6.54	7.00	8.41	8.57	9.80	12.3	14.2	14.6
Mode	9	10	11	12	11	14	15	16
Freq	15.2	16.4	17.8	19.3	22.9	26.1	26.9	30.2

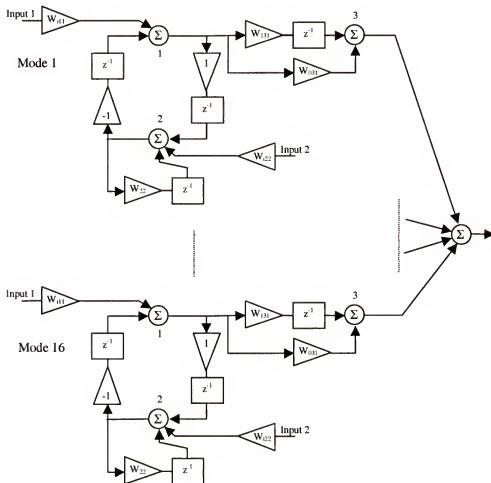


Figure 4-27 Complete 16-mode LLM architecture.

It is reasonable to anticipate that there are practical boundaries defining the minimum and maximum length of signal that a network can be trained to follow. Determining these boundaries is necessary to ensure that an optimal solution is found

through adaptation. As has been shown, oscillatory networks need enough data to fully define the frequency content of the signal, so a minimum amount of data is always required. On the other hand longer signal lengths are desirable since they allow finer spectral resolution. But oscillatory networks are difficult to train for long signal lengths, so it is necessary to establish reasonable limits on the length of training signals used.

Two types of experiments were designed. First, an experiment was set up to evaluate the network's output as the length of the desired response is increased. This increased-time experiment is intended to define an upper bound for the maximum length of the training signal. Prior analysis of the oscillator sub-network indicates that frequency resolution would be difficult with signals one-second in duration or longer. So the experiment was designed to test to a clear failure that was expected to be no longer than five seconds.

The second type of experiment used the entire Test 8c signal as a desired response, subdivided into consecutive, fixed length segments. Networks were trained sequentially, one to each data segment, with the parameters of each initialized to the trained parameters of the previous network. The purpose of these consecutive-time tests was twofold. First to test to a minimum extreme since it may be desirable to implement a network in real time. The idea is that data would be measured and a network trained over a short period of time, then used to predict ahead a few samples, while at the same time recording more data for the next network from which another prediction is made, and so forth. The process would be continuous, so there would always be a near instantaneous prediction of events to occur. Experiments were run with different training signal durations to refine the limits for the necessary 'bursts' of information

The second purpose was to gather information to measure the subtle change in frequency content of the signal over time for use in a parametric evaluation. Changes in each mode can be tracked as functions of time and flight dynamics by evaluating the trained network weights, from which modal frequency, amplitude and phase can be determined.

The two types of experiments were grouped by training signal length:

- Increased-time training segments, such as 0 – 1.0 sec., 0 – 1.5 sec., ..., 0 – 5.0 sec.,
- Consecutive one-second bursts, such as 0 – 1 sec., 1 – 2 sec., etc.,
- Consecutive half-second bursts, such as 0 – 0.5 sec., 0.5 – 1.0 sec., etc.,
- Consecutive quarter-second bursts,
- Consecutive eighth-second bursts,
- Consecutive sixteenth-second bursts.

Since the Test 8c signal shows nonlinear tendencies, it is interesting to look at subtle changes in the frequency content over time. A spectrogram of the Test 8c flight test signal is shown in Figure 4-28. A spectrogram tracks spectral energy over time in a signal, and in this case plots it as different color. Time is shown on the x-axis, frequency on the y. The spectrogram was generated using a 512 point zero-padded fft, a 432 point Hanning window, and a 216 point overlap, in order to give a frequency resolution of about  $\frac{1}{2}$  Hz. Red colors indicate high energy levels (dB), yellow lesser, and blue lesser still. As expected most of the energy is band limited to two frequencies, approximately 7 and 14 Hz, with subtle changes to each occurring over the duration of the signal, especially in the area 14 Hz.

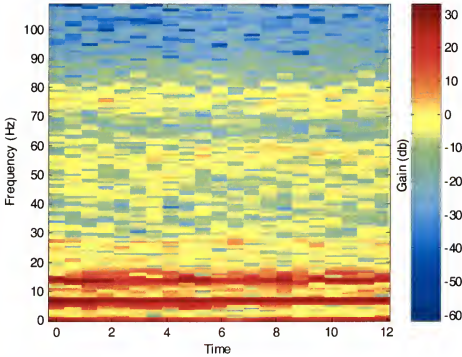


Figure 4-28 Test 8c Spectrogram.

### Evaluation of the Practical Value of MSE for Training

Before experimentation with actual flight test data begins it is necessary to determine a practical working value for the maximum allowable value for the MSE used to determine when a network is sufficiently trained. Although LCO is generally modeled as a deterministic process, when it is combined with measurement noise in the instrumentation, as well as other systematic limiting factors such as inexact and subtly changing flight conditions, LCO can be considered a random process for the purpose of MSE definition. Its distribution is unknown, but since the MSE used in training is a small value, it is sufficient to assume Normal distribution. The variance for a given sample<sup>60</sup> can be defined as

$$\hat{\sigma}_x^2 = \frac{1}{N} \sum_{n=1}^N (e_x - \hat{m})^2. \quad (4-10)$$

If a near-zero sample mean,  $\hat{m} \approx 0$  is further assumed, the variance becomes the same as the MSE

$$MSE(e) = \frac{1}{N} \sum_{n=1}^N (e_x)^2 . \quad (4-11)$$

It will be seen that this is not always a valid assumption, but will be used here in order to develop the training standard MSE. If it is noted that the mean is non-zero, a bias offset can be added to the network architecture.

In order to ensure that a network trains sufficiently to represent the data, while still having some generalization abilities, the maximum allowable MSE value can be viewed as representative of a bound on acceptable error. Suppose an error of  $\pm 0.15g$  from the accelerometer can be tolerated. Statistically, that error can be assured of 99.5% of the time if a standard deviation  $\sigma$  is defined such that

$$3\sigma = 0.15g . \quad (4-12)$$

This equates to a value for the maximum MSE

$$MSE_{\max} = 0.0025. \quad (4-13)$$

Testing showed that this was not always attainable. Many times network training stabilized at MSEs on the order of  $10^{-2}$  corresponding to a  $3\sigma$  value of  $0.3g$ 's. In practice this is still an acceptable error. Therefore it is concluded that for this work the target training MSE values are contained in a window defined as

$$0.01 < MSE = 0.001. \quad (4-14)$$

### Increased-Time Experiment Results

A total of 159 increased-time experimental networks were trained and evaluated. The more significant results will be shown here. Examples of the results are shown in Figures 4-29 – 4-34. These show training over a one-eighth second interval (Figure 4-

29), a one-quarter second interval (Figure 4-30), a one-half second interval (Figure 4-31), a one second interval (Figure 4-32), a four second interval (Figure 4-33), and finally a five second interval (Figure 4-34), which indicates a failure to track.

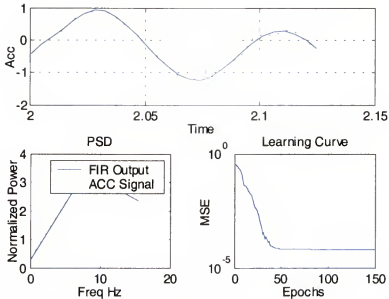


Figure 4-29 One-eighth second signal training test.

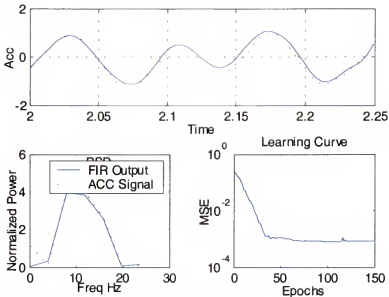


Figure 4-30 One-quarter second signal training test.

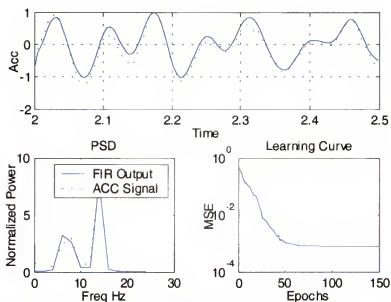


Figure 4-31 One-half second signal training test.

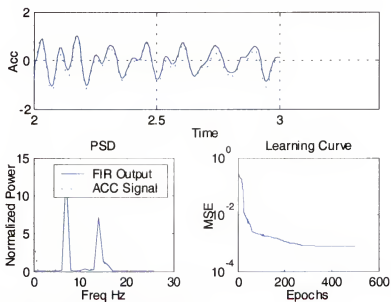


Figure 4-32 One second signal training test.



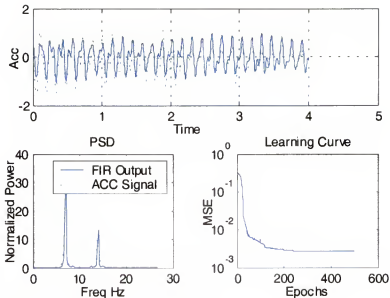


Figure 4-33 Four second signal training test.

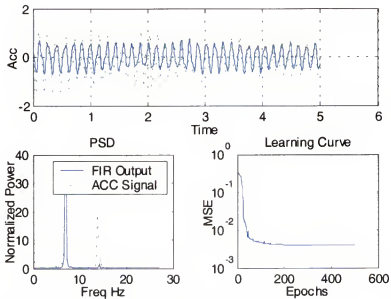


Figure 4-34 Five second signal training test (failure).

The lower bound for training signal burst length seems to be between an eighth and a quarter of a second. The results of a typical eighth-second burst training set are shown in Figure 4-29. The figure contains a signal trace showing the network output

superimposed on the desired response, as well as a PSD of both, and the learning curve produced during training. Notice that learning was rather rapid, that the PSD's are nearly coincidental, and that the network output appears to be coincidental the desired response. Unfortunately this is an indication of over fitting, reducing the network's ability to generalize.

Generalization is an easy concept to understand. It's simply the ability of a network to perform well when new data—information that the network hasn't yet seen—is applied. Unfortunately generalization is difficult to quantify or predict in a neural network. Occam's razor can be modernized to give a guideline to follow: 'Any learning machine should be sufficiently large to solve the problem, but not larger.' When the network is too small it can't sufficiently achieve a good fit to the desired response. If the network is too large, it will not generalize well because the fit is too specific. The network is memorizing the data.<sup>59</sup> In the case of the network trained with very short bursts of data, it appears that there may be too many degrees of freedom to enable good generalization. It is an accepted rule of thumb that the ability of a network to generalize is inversely proportional to the network's degrees of freedom and directly proportional to the size training data set. Since the number of degrees of freedom in the network is constant, generalization should improve as training signal lengths increase. Results from training longer bursts tend to back this up.

Results from a quarter-second burst are shown in Figure 4-30. Notice that the network output and the desired response are no longer quite the same, but that the output follows the trend of the desired response very well. Also, the PSD of the output is very close to that of the desired response, indicating that the network adjusted the frequency content of the data properly. The learning curve shows that the network trained to a

stable solution rapidly, but that the steady-state MSE is higher than that of the eighth-second burst. This is interpreted to mean that an exact fit to the data was not as closely achieved, and therefore better generalization and prediction from this network can be expected.

Figure 4-31 shows results from training a network with a one-half second burst, and those of a one second burst are shown in Figure 4-32. Notice in both figures that the network outputs match the desired response very well, as do the PSDs. Good generalization can be expected from these networks, based on the frequency matching and reasonable mean squared error values. It can be asserted from the results of the one-eighth, one-quarter, one-half and one second experiments that the minimum length of training signal necessary for good generalization is probably no less than one quarter of a second.

Examples of the four and five second training signal experiments are shown in Figures 4-33 and 4-34 respectively. The results of the four second tests show acceptable tracking of the training signal, and an acceptable MSE. Training errors leveled out with MSE values between  $10^{-2}$  and  $10^{-3}$ . The signal trace in Figure 4-34, however, shows that the five second training set is inadequate. The training data displays a signal beating behavior, as well as a decrease in the amplitude noticeable at about 2.5 seconds that the network failed to adequately track. But the network's MSE values stabilized below acceptable levels, normally an indication of successful modeling. This can be explained by again realizing that the mean value of an oscillatory desired response can be near zero. Therefore a network can be trained to low MSE values and still not provide acceptable results. This was the case with in all the five second signal experiments. Especially noticeable is that the network has nearly eliminated all but the 7 Hz primary mode.

These results illustrate that MSE alone is inadequate as a method for determining the applicability a model for a given response. The need for a time-dependent statistical test such as the simplified GLRT is reinforced.

As these experiments were executed it became apparent that initializing the zero-delay weight values for each filter to the mode values given by the flutter solution helped optimize the training process. Recall that not only are the modal frequencies calculated, but the modal amplitude is calculated as well. Originally the filter weights were set to a value of 1.0. Intuitively, though, it made sense to use the values made available by linear flutter analysis. By setting the initial FIR filter weights to match the mode heights, the network has an immediate advantage in conforming to each mode during training. This was done whether the value identified in flutter solution was positive or negative, since a negative value simply means that the phase is initialized at  $180^\circ$  rather than zero. For data taken from the forward wing tip accelerometer mode heights from DMM point 133 were used. When aft wing tip data was used the mode heights were taken from DMM point 132. Modes for both DMM 132 and 133 are given in Table 4-7.

Table 4-7 Initial Modal Amplitudes (flutter solution results).

DMM 132								
Mode	1	2	3	4	5	6	7	8
Amp.	1.0705	-0.5607	-0.0090	-0.2230	0.9780	0.2441	-0.5980	0.0303
Mode	9	10	11	12	11	14	15	16
Amp.	1.0112	-0.0566	1.0426	-0.1831	-0.0831	0.9114	0.1126	-0.6979
DMM 133								
Mode	1	2	3	4	5	6	7	8
Amp.	0.0943	-1.4795	-0.1742	-0.1700	1.1113	0.9445	2.4518	-0.8728
Mode	9	10	11	12	11	14	15	16
Amp.	-2.2759	0.4710	-1.4931	-1.4474	0.0802	-0.2507	-0.4157	-0.1114

#### Consecutive Fixed Length Experiments (Test 8c)

Experiments that evaluate the use of consecutive fixed length bursts of training data were accomplished next. These experiments took advantage of the lessons learned in the

initial Test 8c experiments where it was determined that the most practical method of initializing the frequency controlling and FIR weights ( $W_{22}$  in Figure 4-27) was take advantage of the results of linear flutter analysis methods. The input weights were set to provide a sine wave with unit amplitude. Training was accomplished on the frequency controlling weights, and both filter weights, effectively giving the network three degrees of freedom for each mode in the 16-mode set. This allowed the network to modify the frequency, phase, and gain of each mode. During the course of these experiments these initialization methods were modified slightly as will be discussed.

It is known that linear flutter analysis provides an initial indication of the dynamics of the structure, but doesn't match the measured parameters in flight, and gives no indication as to the onset and severity of LCO or flutter that can be expected. It follows then, that a network initialized to these parameters will most likely not match flight test data and will require modification through backpropagation to change its parameters. This is the basic premise of the proposed paradigm and has been shown by the previous experiments. It further makes sense to use the knowledge gained through training in subsequent networks rather than continually reverting back to the original initial conditions for any subsequent networks used to synthesis a particular data stream. It is also known that the process of LCO is nonlinear, so the use of a constant, fixed-term ANNs is limited to small portions of the flight test signal where parameters remain consistent.

Practically speaking, it has been shown that a single network of the current design should be able to follow a signal until the dynamics change sufficiently for the network to no longer be valid. As an improvement on this, when the network's output begins to diverge from the flight test signal, the network could be retrained to follow the next

portion of signal until such time as it again begins to diverge from the desired response. It makes sense that the trained network parameters of each preceding portion of flight be used as initial conditions for the next network, since it must be assumed that they are closer to matching the state of the required network than those from an off-line analysis. This process, in theory, could then proceed indefinitely. By doing so the result is a time varying ANN that is linear and feedforward in design, at least in a macro sense, but is able to replicate the time-dependent changes of a nonlinear signal. Networks in this study were trained at fixed intervals and the results evaluated. Once it is shown that sequential networks can successfully synthesize a desired response indefinitely, automatic retraining or network switching can be introduced to optimize and automate the process. Dynamics of the system during flight can be further analyzed by evaluating the network parameters and the frequencies, phases and gains they represent. Each mode can be separately and accurately evaluated throughout the entire flight test.

As a starting point, a serial sequence of networks of the type shown in Figure 4-27 were trained to follow the entire 12 second Test 8c signal duration. These sequential networks were trained at fixed times and the results evaluated. Since it appears eighth-second training signal lengths reduce generalization, burst lengths of 0.25 seconds, 1.0 second, 4.0 seconds, and were used. This resulted in 51 sequential networks based on the 0.25 second bursts, 12 for the 1.0 second bursts, and 3 for the 4.0 second bursts.

Sequential networks were trained over the entire 12 second span of Test 8c. The results, regardless of the length of each burst used for training, were very encouraging. The networks were consistently able to match the segmented signal with mean squared error (MSE) values well below  $10^{-2}$ . Most training sessions successfully ended when the desired  $10^{-3}$  MSE value was reached. In general, training the first segment in the

sequence—the first network to be trained in the sequence, initialized with linear flutter analysis results—was the most difficult. When shorter signal bursts were used, the networks usually retrained quickly. The combined learning curves for each training session can be viewed as a measure of the validity of a particular network relative to the next segment in the sequence. That is, training that is quickly accomplished within just a few epochs and relatively low initial value of the MSE at the start of training indicates that the weight values of the previous network are near those required for the current sequence of data. Of course it may also indicate that there may exist a network that is sufficient for both sequences—a topic for further study.

Results for sequential networks will be presented in several figures similar to the initial Test 8c experimental illustrations. Shown in each figure is a signal trace of both the desired response and network output over time in an upper window, the power spectral densities of both overlaid in a window in the lower left of the figure, and a combined learning curve in the lower right corner. The learning curve is sequential. That is, learning curves from training each sequential signal burst are shown from left to right. The end of one burst is usually indicated by either a flat spot in the curve followed by a peak, or a minimization of the curve below  $10^{-3}$ , again followed by a slight peak. The combined learning curve is a good indicator of the validity of previous networks to a current signal burst. An indication that parameters from the previous burst are very near the current parameters is giving if the learning curve peaks only a relatively short distance 'upwards' then quickly converges to a minimum network.

A second figure accompanies the signal trace, PSD and learning curve. This is a comparison of the network parameters. The graphs show the values for all network weights as they change with each subsequent segment. The input weights are shown for

reference only as they are fixed and not trained. The frequency weight is the delayed feedback weight in the oscillator circuits that control frequency. The closer the weight value is to 2, the lower the resulting frequency for that oscillator. The low pass adaptable filter weights are shown as well. Recall that the zero-delay value is initialize to match the mode value given by linear flutter analysis techniques and may be positive or negative. The one-delay weight is initialized to zero, so its values tend to migrate either positively or negatively as the need may be in order for the filter to properly adjust both amplitude and phase.

The final amplitude and phase adjustment for each mode is a combination of the two FIR filter weights given by the relationships

$$\alpha = \left[ W_0^2 + W_1^2 \cos^2(-\omega) \right]^{\frac{1}{2}} + \left[ W_1^2 \sin^2(-\omega) \right]^{\frac{1}{2}} \quad (4-15)$$

$$\phi = \tan^{-1} \left\{ \frac{W_1 \sin(-\omega)}{W_0 + W_1 \cos(-\omega)} \right\}, \quad (4-16)$$

Where  $\alpha$  is the filter gain,  $\phi$  is the filter phase shift, and  $\omega$  is the digital frequency of oscillation.

Each experiment was repeated four times to check the consistency of the training process. Each time the networks were trained they settled to the same values for all weights. It is therefore concluded that for each training set there exists a unique set of network parameters that optimizes the ability of the network to trace that signal, and that the solution curve is such that the parameters can be reached with the modified gradient decent methods used here.

Results for training the Test 8c signal with quarter-second bursts are shown in Figure 4-35. Notice that the training signal and the network output overlay very well on the signal tracing. Peaks of maximum oscillation—the most important part of the signal



since it indicates that maximum force on the aircraft wing—agree. Further, notice that the signal beating is well represented by the network's output. The two PSDs also match. Perhaps most interesting is the composite learning curve. Rises or peaks in the curve indicate changes from one network to the next. Notice that the first quarter-second burst takes quite a while to train and never trains to the desired value of  $10^{-3}$ . It takes three more bursts for the signal to settle to a point where a network can train to the desired minimum. This is an indication of a rapidly changing signal, as can be noted in the signal trace. The first second and a half of the signal in all likelihood still contains some energy from the input of forced oscillation. The first half-second or so contains oscillations that are damping rather rapidly compared to the rest of the signal.

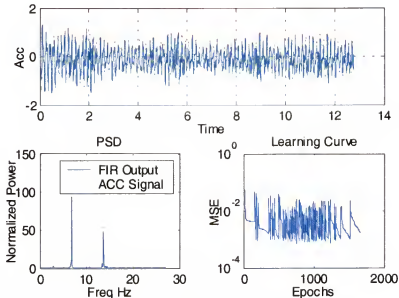


Figure 4-35 Test 8c quarter-second burst training results.

Once the signal has entered an area of relative stability, each subsequent network trains very quickly to its minimum MSE value. Also, there is a period of time, indicated in the learning curve at about 600 epochs, corresponding to about 2.5 seconds in the signal trace, where the maximum error is very small, on the order of  $10^{-2}$ . This implies

that a single network may possibly be used to represent this particular section of signal. This phenomenon is observed a few more times over the course of the signal. Towards the end of the signal it is noted that the signal begins to beat at higher amplitudes again. The learning curve reflects these changes by showing that it is taking longer to adapt to the final sequences of signal.

Figure 4-36 is a detail of the changes in the network parameters. Notice that the frequency weights take a while to settle, especially the values near 2, the low frequency terms that are the most critical for matching the response. They tend to settle down to constant values after the first ten segments, but some change is still noted, especially towards the end. The same is true for the zero- and one-delay filter weights.

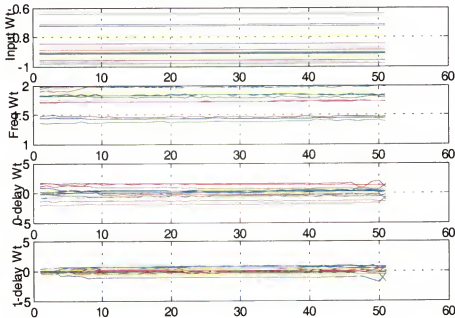


Figure 4-36 Test 8c quarter-second burst network parameters.

The higher gain weights take some time to settle and reach equilibrium, then change rapidly towards the end of the signal, as the signal dynamics again change. It is clear from the figure that often there are only slight changes in network parameters,

implying that longer periods of time could adequately be modeled by a more general network.

Results for one-second bursts are shown in Figure 4-37. This is perhaps the most interesting case to look at since it shows very good overall match, and again gives an indication that there are perhaps regions of the signal that can be serviced by one network for a longer period of time before retraining is required.

These networks successfully replicated the peak amplitudes in the signal, as well as the beating that can be observed in the desired response. The PSDs of both desired response and network outputs line up very well, indicative of proper placement of the majority of the signal power. Evaluation of the learning curves is interesting. Notice that it took three segments of data before the network trained to the desired minimum. This is again indicative of the changing characteristics of the training signal. After the first three segments are adapted the network does train to the minimum, and the epochs required to do so decrease dramatically for a period of time. There is a section in the middle of the learning curve that implies that there is very little difference between one network and the next. More training was required to match the final portions of flight test data. This once again leads to a conclusion that there may exist a series of sequential networks that may match this 12-second portion of flight with fewer than the 12 networks here.

Network parameters for this experiment are shown in Figure 4-38. The changes noted in the frequency weight are subtler, as expected. The frequencies stay relatively constant throughout the signal. The amplitudes and phases seem to change over time indicated by the changes to the filter weights. The higher gain values tend to shift back and forth between the zero- and one-delay weights. This may imply that the overall gain and phases are changing over time rather than the frequency values themselves. It may

also imply that these changes are results of noise in these lower energy modes and not necessarily a good reflection of the mode itself.

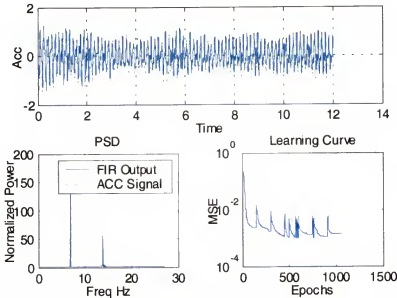


Figure 4-37 Test 8c one-second burst training results.

Again note that there are periods of consistency in the weights. Most noticeable are the periods between 2.5 and 4.0 seconds (the segment number corresponds to time in this plot, since there is one segment per second), and 4.5 to 9.0 seconds. It may be possible to parse this signal into four or five sections and train networks to follow each. This analysis provides further insight as to how automating the training-prediction paradigm will optimize the process.

To further explore the notion of a network's applicability over varying signal durations, an experiment was accomplished that used each sequential network to predict into the time frame of the next in the series. In other words, suppose network 1,  $N_1$  is trained to follow the signal between  $0.0 < T_1 < 1.0$  seconds. But now use  $N_1$  to predict further ahead into the time of  $N_2$ ,  $1.0 < T_2 < 2.0$  seconds and overlay the results on a graph. It would be expected that a successful signal tracing from  $N_1$  would be observed

for the first second, after which it would begin to diverge from the truth in the next second if it was not still applicable.

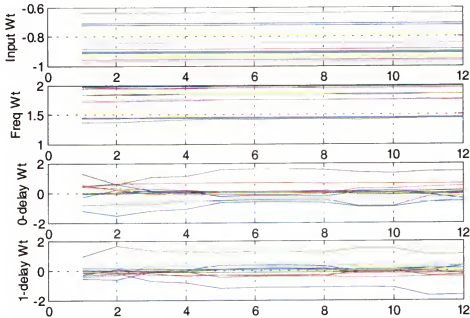


Figure 4-38 Test 8c one-second burst network parameters.

At the same time  $N_2$  successfully models the signal in the second segment. But suppose  $N_1$  and  $N_2$  are nearly identical. The signal tracings would then overlap, at least for the period of time when the two networks were indeed similar. Figure 4-39 is just such a plot, where the output from each preceding network is allowed to ‘bleed’ into the time of the next. The flight test signal is overlaid with connected and overlapping network outputs in the upper window. The second window is a trace of the raw error between each segment’s network output and the actual signal for the corresponding time period. The first few networks don’t predict very far into the next time window, but the third through the eight networks seem to work relatively well with respect to one another. It is interesting to note that there are times when it appears one network may be able to do the job of many, especially between 2.5 and 9.0 seconds. This is nearly the same time

frame that was indicated by observation of the combined learning curves for the same signal/network set.

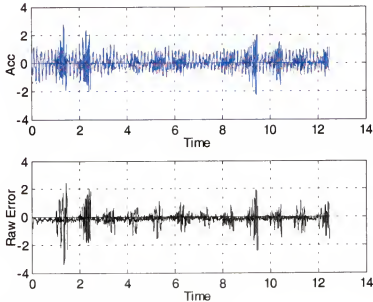


Figure 4-39 Test 8c one-second prediction plots.

The concept that fewer networks are required for this portion of flight test is further reinforced by the results of training just three networks to model four seconds of data each over the length of the Test 8c signal. The results of that study are shown in Figure 4-40. Again, the flight test signal can be traced with the network outputs very closely, and the PSD's line up nearly identically, as is now the expected result. The learning curve once again provides insight into the system.

The beginning network never converges to a solution whose MSE was below the desired value of  $10^{-3}$ , but is still able to drop well below  $10^{-2}$  and thus provide a very reasonable signal match. However, the MSE for the next segment didn't rise too far above  $10^{-2}$ , indicating that the network parameters for both were in neighborhood of each other. The same is true for the transition from segment two to segment three, although the shape of the two learning curves are different enough to suspect a difference in

required networks as has become the expectation since the signal characteristics change near the end of the sequence.

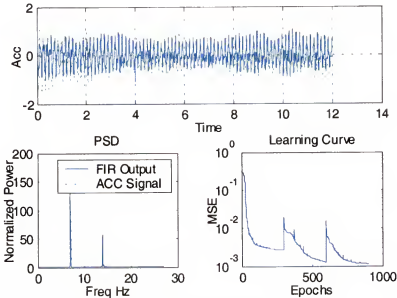


Figure 4-40 Test 8c four-second burst training results.

Closer inspection of the signal traces gives an indication that the four-second window may be too wide to adequately match the first segment of the signal. Energy from the oscillation injected by the flutter excitation system on the aircraft has been noted in the first second or so of the signal. During this part of flight, the network output is no longer following the peaks as closely as desired, and the network would subjectively be deemed a failure since it no longer predicted large enough amplitudes in the oscillation. Some of the beating recognized in the flight test signal is more subdued in the network outputs. This is a further indication that at least for this portion of signal the fixed valued of four seconds for each segment is too long.

Some insight is gained from this experiment, however. When the network parameters are evaluated over the three sets, it is seen that the frequency values remain nearly constant. The changing parameters are the two filter weights that account for the

overall amplitude and phase of each frequency in the signal. Notice in Figure 4-41 that there emerge two primary modes that change over time, while the other modes seem to be relatively stable to the each other. These are the two modes containing the primary energy in the PSD. The other modes change more subtly to fill out the PSD as required.

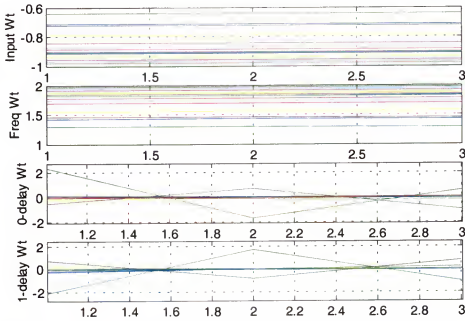


Figure 4-41 Test 8c four-second burst network parameters.

The results of these experiments lead to the conclusion that a flight test signal of significant length can be successfully synthesized with a series of sequential neural networks. These networks can be valid for long periods of time relative to the changing dynamics of the underlying process. By training and connecting sequential linear networks a system can be constructed capable of simulating and predicting a nonlinear, time-dependant system. Further, the network's components can be dissected in order to gain insight into how each mode in the system reacts during flight. Changes in frequency, amplitude, and phase of each mode can be tracked as a function of time. This



has been shown by matching the signals from a 12-second portion of flight containing moderate LCO over a steady set of flight conditions. Further examination of the system can now be made by introducing it to a longer, much more dynamic portion of flight test data.

### **Consecutive Fixed Length Experiments (Test Line 17)**

Test line 17 provides a long signal with interesting behavior for exercising and refining the LLM data synthesis paradigm. Figure 4-42 shows data from the left wing forward wing tip accelerometer (repeated here from Chapter 2 for clarity). Limit cycle oscillation is already present at low but sustained levels in the beginning of the segment. The amplitude of the oscillation increases steadily with increasing Mach number and appears to be flutter at first glance. But a close inspection shows that as Mach number becomes stable around 0.9, the oscillation amplitudes stabilize as well. The last five seconds are steady, beating limit cycle oscillations with large maximum amplitudes around 3.5 g's. The PSD for this signal is also shown in Figure 4-42. Once again the 14 Hz sub-component is present in the signal although at relatively lower energy than in Test 8c. Noteworthy in this signal is that there is a distinct bias in the negative g direction. Also, as pointed out in the detailed discussion of this data set in Chapter 2, the flight conditions can no longer be considered as constants, since as the amplitude of LCO grows it is evident from the  $N_z$  trace that the pilot is experiencing difficulty in maintaining steady normal acceleration. The test was terminated shortly after this signal was recorded due to high amplitude, sustained LCO. This seemed to be a good data set to challenge the current architecture and sequential neural network process.

Since results from Test 8c gave the most information when one-second windows were used to divide the training signal into segments, the same process was followed with

this new data set. The current network architecture (Figure 4-27) was reinitialized as before with frequencies set to free vibration values and zero-delay weights to modal linear flutter analysis amplitudes. One-delay weights were again set to zero. Training began with the maximum number of epochs per network set at 150, and the desired MSE value set at  $10^{-3}$ . Results from the first attempt are shown in Figure 4-43.

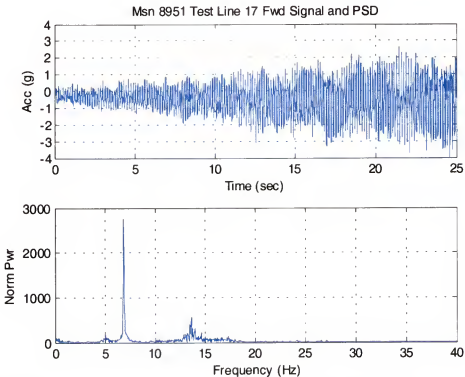


Figure 4-42 Test line 17 forward wing tip accelerometer signal and PSD.

The results are no longer consistent over the entire signal; that is the networks were successful in matching some portions of signal, but not all. Further, the bias played a large role in the system's failure to match the entire signal. The network output continues to oscillate around zero mean, but the flight test signal does not. The problem gets worse as the amplitude of oscillation increases, as indicated by the learning curve's trend upward.

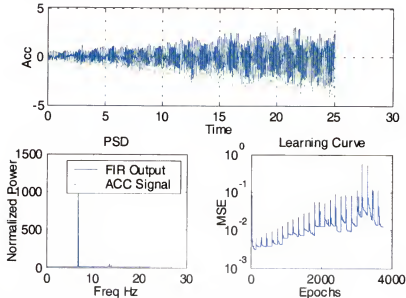


Figure 4-43 Test line 17 original architecture training results.

The improvement to the networks' ability to model the signal is immediately apparent from the figure, as once again the learning curve is indicative of the networks' behavior. The initial network trained to the desired MSE very rapidly, and subsequent networks were able to maintain similar MSE values for some time. As the dynamics of the system increase the networks again begin to struggle to match the signal. The last six networks trained have difficulty training to reasonable values. The time in flight for these networks is coincidental with when Mach number begins to stabilize at 0.9, and the oscillations are transitioning from increasing amplitude to true LCO at very high amplitudes.

Network parameters are shown in Figure 4-44. Although it is apparent that the architecture is inadequate, there is some insight to be gained from a parametric evaluation. The FIR filter weights are changing quite rapidly from one model to the next. One explanation for this is that the model is attempting to adjust the phase and gains of

each mode to accommodate the system bias. The obvious solution if this is the case is to add a bias term to the output of the model.

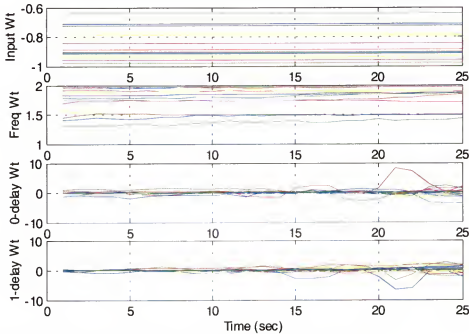


Figure 4-44 Test line 17 original architecture network parameters.

The problem may also be that the dynamics of the system are changing too rapidly to be followed by networks trained every second. There are two possible solutions that may reduce this effect. First, networks could be trained at closer intervals in an attempt to keep up changing system dynamics. It has already been shown that networks can be trained to follow the signals for periods of time as short as an eighth of a second. Decreasing the desired signal length is not necessarily an attractive solution, though. It has been shown that the simplified GLRT requires an absolute minimum of 70 data samples to establish stable statistics. Better statistics, and therefore better signal change detection is done with the most data available. An alternative may be to increase the memory available to the network. This can be done by simply increasing the number of

delays in the filter stage from one to two. Stated a different way, by adding a delay and weight value to each oscillator filter stage, an additional degree of freedom from which the network can work is made available to form the optimal solution.

Figure 4-45 shows the new network architecture with the addition of a second delay in the linear filter stages, and a bias term at the network output.

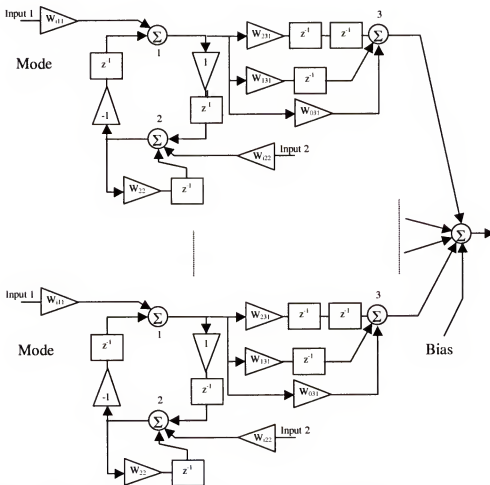


Figure 4-45 Complete two-delay oscillator architecture with output bias.

This architecture was used, again with the weights for the higher order delays initialized to zero, and the results are shown in Figure 4-46. Notice now that the entire

flight test signal has been successfully traced from beginning to end. Further, each network trains to acceptable MSE values.

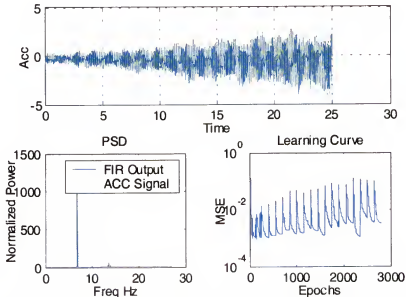


Figure 4-46 Test line 17 two-delay, biased architecture training results.

Figure 4-47 shows the network parameters as they change over time. As was the case in with the Test 8c experiments, the frequency values stay relatively constant. There is some time-dependence of the frequency weights at higher frequencies, but their gains are very low. They may contribute to some subtle shaping of the network output, or may be attributed to noise. It is interesting to notice that for the first eight seconds the filter weights are nearly constant, and the second-order weights are very small compared to the other two. They begin to be more significant at about eight seconds and continue to actively change up till the end of the training signal. At about ten seconds all three filter weights are beginning to actively change with time as the accelerometer signal transitions from increasing amplitude oscillation to bounded LCO. Although an argument can be made that similar results would have been observed if shorter signal bursts had been

used, it is interesting to note that the network took advantage of the additional degree of freedom only when it appeared necessary.

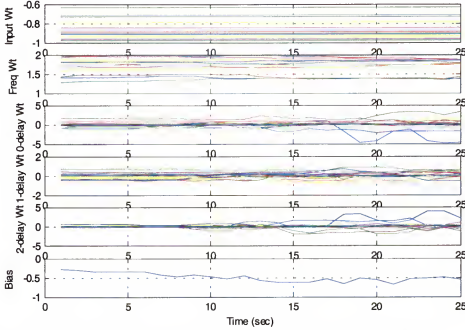


Figure 4-47 Test line 17 two-delay, biased architecture network parameters.

The bias is a very active time-dependent variable. Its value quickly moves from zero to a significant value immediately after training began. It is obvious that the adaptation process took advantage of the bias term to ensure a complete signal match.

The filter frequency responses can again be calculated if desired. Equations 4-15 and 4-16 must be modified to account for the extra degree of freedom, but the form of the equations remains consistent.

$$\alpha = \left[ W_0 + W_1 \cos(-\omega) + W_2 \cos(-2\omega) \right]^2 + \left[ W_1 \sin(-\omega) + W_2 \sin(-2\omega) \right]^2 \Bigg|^{1/2} \quad (4-17)$$

$$\phi = \tan^{-1} \left\{ \frac{W_1 \sin(-\omega) + W_2 \sin(-2\omega)}{W_0 + W_1 \cos(-\omega) + W_2 \sin(-2\omega)} \right\}. \quad (4-18)$$

## Chapter Summary

The goal of this chapter was to expand the basic tools necessary to implement a working data synthesis algorithm from their formal theory to details of their practical abilities and limitations. The two major components of the paradigm, the LLM network model, consisting of the linear oscillator and the FIR filters, and the simplified GLRT, were evaluated and refined through experimentation and analysis.

It was shown that the linear oscillators designed in Chapter 3 adapt to frequencies best when their frequency-controlling weight is initialized to a value in the neighborhood of the actual value. Adaptation when the weight is initialized at values further from the desired value is more difficult because of the shape of the solution surface. This is not as severe a limitation as it may at first seem since an initial guess for frequency contents of modal systems can be obtained through simple linear approximations, even if those approximations fail to adequately model a nonlinear, time-varying system. In the case of thin wing LCO prediction, free vibration frequency analysis is sufficient to initialize modal frequencies for successful adaptation.

A method for training recursive networks, specifically the linear oscillator, was tested and refined. The RPROP method of fixed-value gradient decent, modified for backpropagation through time was evaluated. The details of the parameters used for training such as step sizes and change-deltas were determined.

The applicability of simplified GLRT as a tool for detecting a change in a modal signal sufficient to warrant the need for a change in the LLM was evaluated through a sequence of experiments. It was determined the simplified GLRT is a viable detecting method and that it can detect changes in a signal if enough data is present to calculate stable statistics. It was determined that although simplified GLRT detected signal



changes in very short periods of time, it is necessary to give the algorithm about 70 samples as a minimum, or about 0.40 seconds of data sampled at rates used for flutter flight testing.

The simplified GLRT was able to identify the need for a model change not only in finite, discretely changing desired responses, but also in continuously changing responses within a given set of minimum errors. This was very encouraging for this work since data has been introduced displaying LCO tendencies that seem to constantly change.

Finally, experiments were conducted that evaluated a specific network architecture that combined the recursive linear oscillators with adaptive FIR filters for modal phase and gain adjustments. These experiments showed that a series of linear neural networks, trained sequentially over a segment of flight test data can successfully synthesize a nonlinear, time dependent system. This was demonstrated initially by using first-order linear FIR filters fed by recursive linear oscillator circuits to replicate a 12-second accelerometer signal having characteristics of typical LCO. Three successively larger signal burst lengths were used to train the sequential networks: 0.25 seconds; 1.0 seconds; and 4.0 seconds. It was shown that the 1.0 second burst were the most versatile, but evidence was presented indicating that many times the signal's characteristics can sufficiently be realized with one network over a longer period of time. These results reinforced the conclusion that a signal change detection method such as simplified GLRT could help optimize the data synthesis paradigm.

As a test of the robustness of the process, a second flight test signal was evaluated using the longer Test line 17 signal. The signal contains rapidly increasing oscillations, finally stabilizing to high energy LCO. The architecture originally proposed needed modification in order to match this signal. A bias was added, as well as the addition of a

second delay in the linear FIR filter for each mode of oscillation. Increasing the filter order from one to two increased the system's memory and enabled it to follow the more rapid changes in the signal. This means that a given network will be valid for a longer period of time before retraining is necessary. It was demonstrated that with this expansion of the network architecture the new flight test signal could be successfully replicated over its entire 25-second duration.

Analysis of network parameters was provided that showed the time-dependency of many of the parameters over the course of a long flight test signal. This was especially evident in the second signal from Test line 17. The bias term was immediately effective in increasing the system's ability to match the signal. Also interesting to note was the fact that the second order filter term was only activated by the training process when the dynamics of the training signal made it necessary to do so. This is in part due to its initialization to zero, making it benign at the beginning of a training session, but demonstrates the ability of gradient decent methods to correctly select and adapt parameters in an architecture properly constructed to match the system.

Finally, an analysis of the completed network was done to calculate its transfer function for an arbitrary filter order. This showed that the system is IIR in nature, but that the poles can be independently placed as necessary during adaptation, increasing the ability of the network to adapt to any portion of flight.

## CHAPTER 5

### FLIGHT TEST EXPERIMENTS

The paradigm for synthesis and analysis of accelerometer data has now been formulated, and practical working parameters and limitations have been defined in the previous chapter. A complete evaluation of the test signals from Test lines 3 and 17 can now be accomplished. This chapter will present examples of experimentation with both sets of data, using the now familiar two-delay network architecture, shown again in Figure 5-1. Test line 17 will be presented first as an example of how the paradigm can be used as an on-line tool for segmenting, modeling, and predicting a signal as it is occurring, depending, of course, on available hardware. Test line 3 will be used as an example of using the paradigm for post processing when the signals change dramatically and chaotically between test conditions and on-line processing may be impractical.

Network training results will be presented, along with analysis of the network parameters. The questions of network repeatability and predictability will be addressed. During experimentation there was some redundancy in the network weights that may indicate that the degrees of freedom in the network architecture may be reduced. A method by which the trained network's FIR weights can be used to reduce the filter's size from a second-order to a first-order filter is proposed and the results given. It will be shown that, as expected, the second-order network architecture is very good at following the accelerometer signals, and that the order of the filters can indeed be reduced with no loss in signal resolution. This chapter will also give an analysis of the modal content of



was accomplished on all networks using RPROP modified for BPTT as detailed in the previous chapter. All networks were trained at least three times in order to ensure consistent results. The actual time required for training each LLM was not tracked. These experiments were accomplished using Matlab® m-files and as such were not optimized for execution speed. It is understood that a real-time implementation of this paradigm would require high-speed hardware running optimally compiled code since the number of operations is very high. Most LLMs trained within 150 epochs. In most cases the time to detect a signal change was greater than the time required to modify the LLM.

### **Test Line 17 LLM Experiments**

Data evaluation begins with the signal from the forward left wing tip accelerometer taken during Test line 17. Recall that this is a portion of flight where, at least from a practical point of view, all parameters remain nearly constant except Mach number. The algorithm was given only a single, initial starting network. The simplified GLRT was used to detect signal changes. When a signal change was detected the network was retrained with weights set at the previous values. Figure 5-2 shows the results of training for the entire signal duration. The MSE value for the trajectory is fairly consistent throughout the entire signal. A maximum MSE value of  $10^{-3}$  was the training goal and was consistently reached in the training sets. As can be seen in the lower window, the overall error and MSE is satisfactory, but higher than the desired value because of the errors at switch times. Overall, this family of LLM networks synthesized the entire signal very well.

The PSD for the combined network outputs is superimposed on that of the desired response in Figure 5-3. Not only is the primary mode correctly represented in the signal by all networks, the PSD of the secondary modes matches the desired response nearly

exactly. Slight differences are noted in the energy in the 14Hz frequency band. The energy is distributed across two of the discrete modal frequencies rather than as a continuous distribution, as would be expected from this type of system.

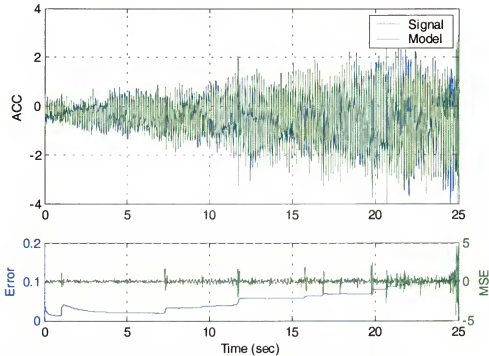


Figure 5-2 Results of Test line 17 auto-switched LLM data synthesis.

The L-values from the simplified GLRT, shown in Figure 5-4, present interesting experimental results. Several observations can be made. Most notable are the even distributions of the time between signal changes. The preliminary paradigm experiments had indicated that a single network should be capable of modeling the signal for about one second. This observation is confirmed in the figure. The time between signal changes was consistently a fraction of a second greater than one second. A total of 25 models were needed to successfully synthesis this signal. There are a few exceptions to the one-second duration. The time between the twelfth and thirteenth models was slightly less than one second, as were the changes between the nineteenth and twentieth

and twenty second and twenty third models. These correspond to portions of the desired response with slightly increased dynamics.

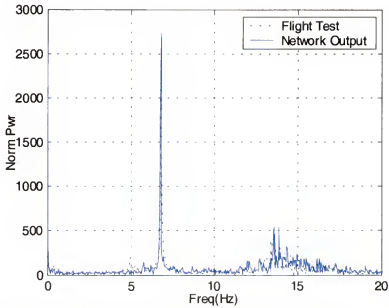


Figure 5-3 PSD for test line 17 auto-switched LLM data synthesis.

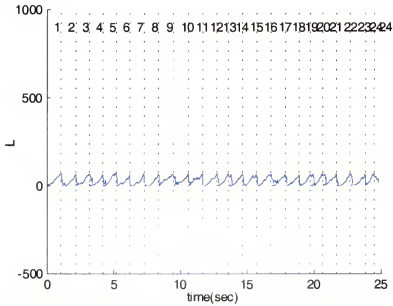


Figure 5-4 L-values for test line 17 auto-switched LLM data synthesis.

Also noteworthy in the figure is the consistency of the L-values from signal change to signal change, and the nearly uniform shape of the L-value curves. Since the flight test signal changes very smoothly as the Mach number increases, the LLMs change at a similar rate in the stair-step fashion noted in chapter 4. So it is expected that the L-values would be consistently low from segment to segment.

### **Parametric Predictions.**

In order to fully utilize the capability of the paradigm it is desirable predict the network parameters for the next LLM from preceding networks and independent variables such as flight test states. An evaluation of the network parameters as they change from LLM to LLM can provide insight into a possible parametric prediction method.

Parametric analysis begins by looking at the change of the zero-order weight as the LLMs change with the desired response. Changes to the weight, labeled  $W_1$ , are shown for mode 1 in Figure 5-5 as a function of Mach number. The weights change in an oscillatory manner with increasing absolute values. The increasing magnitude of the weights is expected due to the nature of the signal. The oscillations in the weight values gives an indication that the weights may be influenced by modal characteristics.

Figure 5-6 gives further insight into the behavior of the weights and the characteristics of the LLMs. The zero-order weights are plotted against the corresponding second-order weights for each LLM. Mode 1 is again used as an example. A very linear relationship between these two weights is displayed. In other words, one of the weights can be modeled as a linear function of the other, indicating that a redundancy may exist in the LLM.



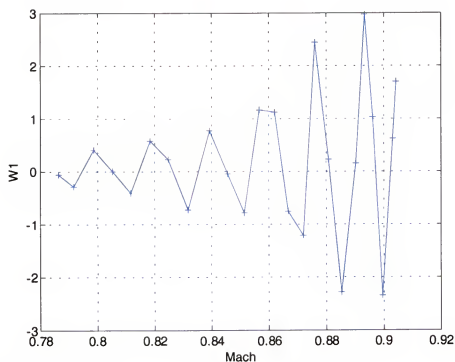


Figure 5-5 Test line 17 zero-order FIR weight (mode 1) vs. Mach.

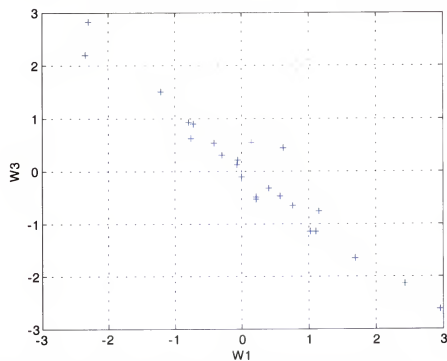


Figure 5-6 Test line 17 zero-order vs. second-order FIR weights (mode 1).

The first-order weights for mode 1 are plotted next in Figure 5-7 where an oscillatory relationship with increasing magnitude is again observed as a function of Mach. Evaluations of the first-order weights with respect to the zero- and second-order weights were made with no linear relationships noted.

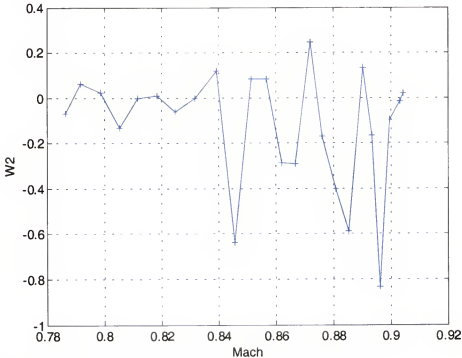


Figure 5-7 Test line 17 first order FIR weight (mode 1) vs. Mach.

Figure 5-8 is a plot of the LLM bias as a function of Mach number. A clearly linear relationship between Mach number and the bias is seen. This observation gives the first evidence that LLM parameters can be predicted as a function of past values and flight test states, at least when only one state is changing, and the change is at least piecewise continuous. Noteworthy in the figure is that as the desired response becomes more complex, the residual errors from a linear approximation of the bias become larger. It can be argued that perhaps a higher order approximation may be required as the

response becomes more dynamic, but no clear form of that approximation can be determined from this data set.

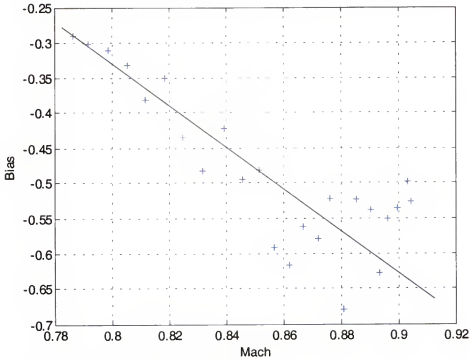


Figure 5-8 Test line 17 LLM bias vs. Mach.

### Reduced-Order Networks

The possible network redundancy noted in the W1 vs. W3 plot (Figure 5-6) can be addressed by an evaluation of the FIR weight vectors. The rule of thumb for a network's ability to generalize over a solution space is that generalization increases as the network size for a given desired response decreases. This, along with the decreased memory requirements and increased training speed, makes reducing the FIR filter length from three weights to two desirable if a redundancy does in fact exist. Moreover, if the LLM is a good model, its parameters would correspond to physical qualities.

There are several possible methods for reducing the size of the FIR filter from three weights to two (any further reduction would inhibit the filter's ability to adjust phase).

The linear relationship between  $W_1$  and  $W_3$  could be exploited with a curve fit, but that would be an approximation and would require evaluation of the relationship between the weights each time a network is trained. A more general solution can be found by realizing that the FIR filters used in the LLM effect only one frequency each. This can be exploited because for each frequency the FIR filter is in effect changing the magnitude and phase of the sinusoid. The FIR filter can be cast as an equivalent phasor, a complex function with a specific gain and phase for its frequency, by substituting  $e^{j\omega}$  for the complex variable  $z$  in the filter's transfer function.<sup>48 p. 253</sup> The same, of course, can be said of the desired two-weight filter. It too must adjust the phase and gain of a specific frequency in exactly the same way as its three-weight counterpart. So, the three-weight FIR filter can be reduced to a two-weight filter by converting its difference equation to a complex weighted vector, calculating the phasor for its specific frequency, and forcing the phasor of a two-weight FIR to be equal to it. Since the phasors are set to be equal, their real and imaginary parts must also be equal. Equations for reduced-order weights can be defined in terms of the real and imaginary parts of the three- and two-weight FIR phasors. Let

- $W_{3R}$      $\equiv$  The real part of the second-order FIR,
- $W_{3I}$      $\equiv$  The real imaginary of the second-order FIR,
- $W_{2R}$      $\equiv$  The real part of the first-order FIR,
- $W_{2I}$      $\equiv$  The imaginary part of the first-order FIR,
- $W_{ij}$      $\equiv$  The  $j^{\text{th}}$ -weight of the  $(i-1)^{\text{th}}$ -order FIR filter.

The difference equation for an  $N^{\text{th}}$ -order filter is

$$h(n) = W_1 + W_2 x(n-1) + W_3 x(n-2) + \dots + W_{n+1} x(n-N). \quad (5-1)$$

Its transfer function is

$$H(z) = W_1 + W_2 z^{-1} + W_3 z^{-2} + \dots + W_{n+1} z^{-N}. \quad (5-2)$$

Substituting the complex variables, its frequency response to a digital frequency  $\omega$  is

$$H(e^{j\omega}) = W_1 + W_2 e^{-j\omega} + W_3 e^{-j2\omega} + \dots + W_{n+1} e^{-jN\omega}. \quad (5-3)$$

An equivalent complex vector representation of the weights can be expressed as a complex exponential

$$W = \alpha e^{j\theta}. \quad (5-4)$$

The real and imaginary parts of the original second-order weight vector and its corresponding first-order vector must be equal

$$W_{3R} = W_{2R}; \quad W_{3I} = W_{2I}. \quad (5-5)$$

The real part of the second-order vector is given by

$$W_{3R} = W_{30} + W_{31} \cos(-\omega) + W_{32} \cos(-2\omega), \quad (5-6)$$

and the imaginary part given by

$$W_{3I} = W_{31} \sin(-\omega) + W_{32} \sin(-2\omega). \quad (5-7)$$

Similarly, the real part of the first-order vector is given by

$$W_{2R} = W_{20} + W_{21} \cos(-\omega), \quad (5-8)$$

and the imaginary part by

$$W_{2I} = W_{21} \sin(-\omega). \quad (5-9)$$

Using Eq. 5-5, the reduced-order one-delay weight can be determined by

$$W_{21} = W_{31} + 2W_{32} \cos(-\omega), \quad (5-10)$$

and the zero-delay weight by

$$W_{20} = W_{30} - W_{32}. \quad (5-11)$$

The conversion to a first order FIR filter from the original second order filter is now complete.

Since it was noted that some of the network parameters appeared to be functions of independent variables (flight states) it may be possible to use the complex form of the

FIR filter vectors to approximate and perhaps predict the network parameters. By looking again at the real and imaginary parts of weight vectors

$$W_R = \alpha \cos(\phi); \quad W_I = \alpha \sin(\phi) \quad (5-12)$$

it is noted that the reduced order weights can be expressed as functions of the filter gain  $\alpha$ , the filter phase  $\phi$ , and the modal frequency  $\omega$ . The one-delay weight is given by

$$W_{21} = \alpha \frac{\sin(\phi)}{\sin(-\omega)}, \quad (5-13)$$

and the zero-delay weight by

$$W_{20} = \alpha \cos(\phi) - W_{21} \cos(-\omega). \quad (5-14)$$

Expressed compactly as

$$W_{21} = f(\alpha, \phi, \omega); \quad W_{20} = f(\alpha, \phi, \omega, W_{21}). \quad (5-15)$$

This means that the weights are themselves functions of components of the complex phasor vectors that may be simply modeled by flight state variables. In other words, a change of an external parameter can be mapped one-to-one with a system parameter; a characteristic of a physical model as was expected. The result is that weight predictions can be made a priori.

The phase angle  $\phi$  is examined for all modes in the entire sequence in Figure 5-9. There are a total of 400  $\phi$  values corresponding to the 25 LLMs, each having 16 modal oscillators. The figure shows the angles grouped by mode so that the linear relationship of  $\phi$  for each mode can clearly be observed. For instance, the angles for mode one are the first 25 plotted, those for mode two are 26 through 50, and so forth. The behavior of modes that are active, such as modes 1 and 2 is very linear. Modes that are less active, those don't contribute to the majority of the energy in the signal, have angular responses

that are less predictable and more chaotic, but it can be said that all angles could be approximated well with a linear fit to their values.

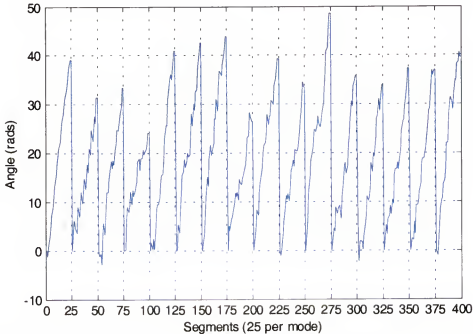


Figure 5-9 Test line 17 reduced-order LLM FIR modal vector angles by segments.

The ability to model the FIR filter angles as a linear function of an independent variable, in this case Mach, is a step forward towards the goal of predicting network parameters. As Eq. 5-15 shows, the only parameter left unknown is the magnitude of the frequency response of the filter. These magnitudes are plotted in Figure 5-10 for each mode as a function of Mach number. Again there is a clear linear trend for the active modes. Active modes can be identified by the magnitude of the frequency response compared to other modes in the set. Modes 1 and 2, the modes in the region of the 7 Hz energy band, as well as modes 7 and 8 in the 14 Hz band have magnitudes much greater than the remaining modes. The magnitude trends of these modes, especially modes 1 and 2, show that they can be approximated linearly. Assuming that this is the case, that these

parameters can be modeled as linear functions of Mach, both FIR filter weights for each mode can now be modeled and predicted.

Figure 5-10 may also provide some further insight into the nature of the LCO phenomenon present in this test. It was noted in the discussion of the flight test data that the modal behavior of the signal in the 14Hz band may be the result of an harmonic of the primary 7Hz flutter mode. The behavior of modes 7 and 8 in the gain plots of Figure 5-10 may provide some anecdotal evidence to support the conclusion that the behavior is indeed due to an harmonic. The gains of their filters are quite high, but oscillate quite a bit. It is true that their trend is a linearly increasing magnitude, but clearly the residual errors from such a fit would be larger than, say, those of a fit to the mode 1 or 2 magnitudes. Compare the behavior of the mode 1 magnitudes to the mode 7 magnitudes. Notice that many times that when the magnitude of the lower frequency increases, the magnitude of the higher frequency decreases. They appear to be oscillating out of phase with each other throughout the different models. Although the frequency content of both modes is real, their behavior seems to be related, indicating that the higher frequency is an harmonic of the lower.

Equation 5-15 also provides a relationship between the zero- and first-order weights. Figure 5-11 shows these weights plotted one as a function of the other. It is clearly seen that when modes are active the relationship between these weights is well behaved. When the modes are not active the relationship is rather random by comparison. Figure 5-11 shows that the weights for modes 1 and 2 have much higher values than those corresponding to the remaining modes, as is expected. Modes 3 - 6 show tight clustering around zero, indicating that for most of the LLMs the adaptive filters are blocking the modal energy. This is also true for modes 10 through 16. The WI



vs. W2 plots for modes 7 and 8 show a wider linear spread, again indicating that contribution of these modes is greater more of the LLMs. The spread of the weights for mode 9 is wider than those of succeeding modes but the relationship is less linear, indicative of a more random energy content rather than a truly active mode. Based on observations of both Figure 5-10 and 5-11 it can be concluded that the most active modes for Test line 17 are 1 and 2, and 7 and 8.

As further evaluation for the selection of active modes and the ability to model LLM parameters, polar plots of the  $\alpha$  and  $\phi$  values are shown in Figure 5-12. These plots give the clearest indication yet of the active modes by virtue of the total behavior of the FIR weight vectors. As an example compare the mode 2 and mode 3 plots. The behavior of the mode 2 filter is very circular and well organized, while that of mode 3' s is random. This circular trend can be seen for modes 1, 2 and 7, while the rest of the modes show haphazard, random behavior. It can be concluded from these plots that these three modes are the most active in desired response.

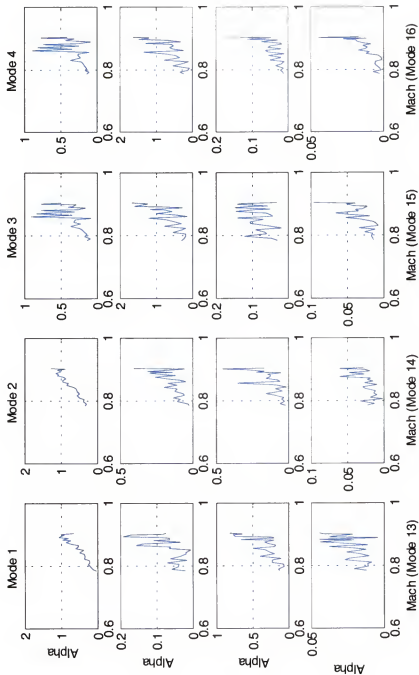


Figure 5-10 Test line 17 reduced-order FIR weight vector angles vs. Mach.

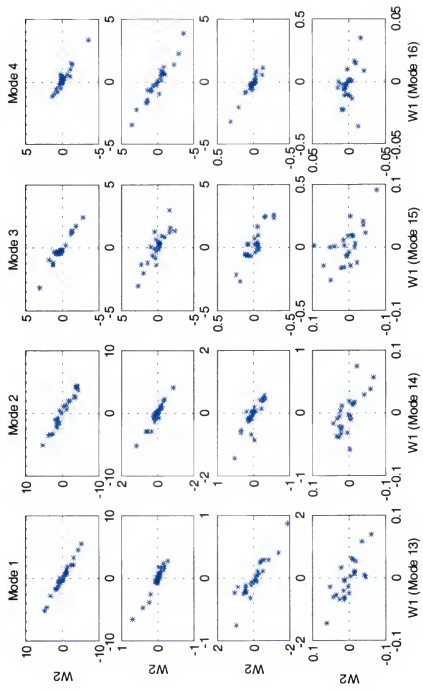


Figure 5-11 Test line 17 reduced-order FIR zero-order vs. first-order weights.



Figure 5-12 Test line 17 reduced-order FIR weight vectors.

### **Reduced-Order Model Results**

The results of the Chapter 4 fixed-length experiments implied that a second order FIR filter may be necessary to successfully synthesize the flight test data. Yet, results of segmentation modeling now imply that this conclusion was false and that first-order FIR filters may be sufficient. Looking back at the fixed-length experiments shows that the conclusion may have been incorrect. The filter order was increased at the same time that the bias was added to the LLM architecture. The influence of the bias may have made the true improvement to the network's modeling ability. Further evidence that this is indeed the case by noting that during the experiments the second-order weights were only used as the dynamics of the system became quite active. This may be an indication that a redundancy does in fact exist and that the second-order weights were used simply because they were available to the training process. In order to further evaluate whether or not a redundancy exists, and if a first-order FIR filter would be sufficient, the adapted FIR filters for the segmented data are now reduced and evaluated.

Reduced-order FIR filters for all 25 signal segments were calculated based on the above method, signals were generated, and the results compared to the desired response. Figure 5-13 shows the new reduced-order LLM sequential outputs along with the flight test data. Comparing Figures 5-2 and 5-3 to 5-13 and 5-14 (the reduced-order sequential output PSD) shows absolutely no change to the outcome of the experiment.

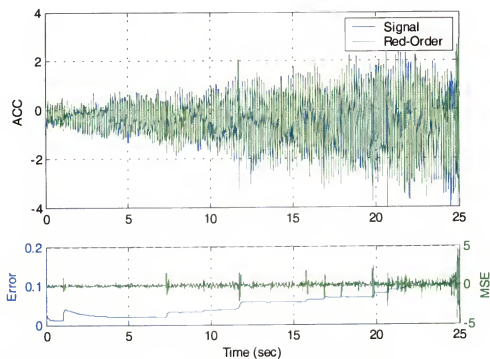


Figure 5-13 Test line 17 synthesis using reduced-order FIR filters.

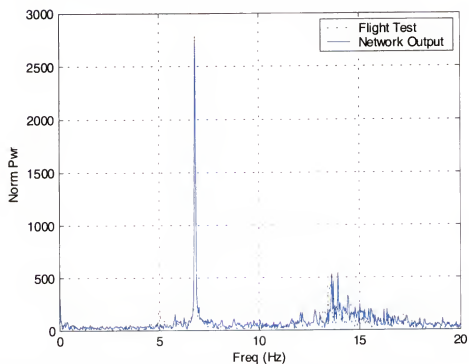


Figure 5-14 PSD of test line 17 synthesis using reduced-order FIR filters.

### Reduced-Mode Results.

Since it has been hypothesized that there are only three active modes in this data sequence, and that they can be identified through parametric evaluation of the LLMs, it is interesting to see if the output of a three-mode set of LLMs can adequately approximate the desired response. The accelerometer data and the output of a three-mode sequence of LLMs are plotted in the first and second windows respectively of Figure 5-15.

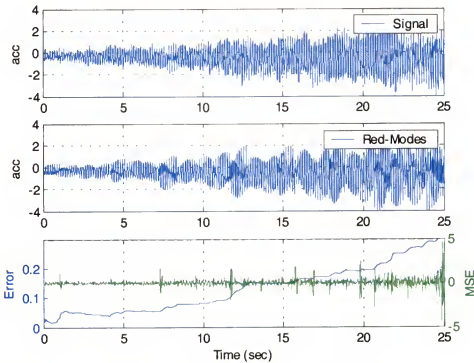


Figure 5-15 Data synthesis of test line 17 using LLMs for modes 1, 2, and 7.

Several interesting observations can be made. The envelope of the desired response is represented very well by the tri-modal approximation, although it tends to exaggerate slightly the more dramatic peaks in the data. These exaggerations are acceptable, however, since from the aspect of safety of flight they represent a realistic but conservative estimation to the signal. Also, the higher frequency noise has been eliminated while the beating trend in the data is retained. This is of course due to the

limited modal data available for the simulation, but the results provide a clearer picture of the important aspects of the desired response. The MSE stay low throughout the entire trajectory, though higher than that of outputs from the full LLM family of models.

Overall, the tri-modal approximation is a reasonable representation of the accelerometer.

It is also interesting to look at the PSD of the tri-modal approximation in Figure 5-16. The primary modes are represented correctly as is the secondary mode in the 14 Hz band. But notice that although there is only one mode contained in the model in the vicinity of 14 Hz, there is more than one frequency 'spike' in the band. Switching between models at irregular intervals, in this case by using the simplified GLRT, can be interpreted by digital frequency measuring methods such as the discrete time Fourier transform as additional frequency content in the signal.

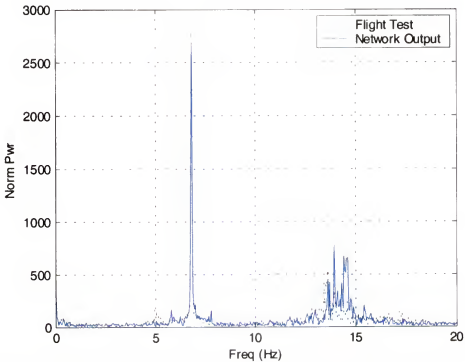


Figure 5-16 PSD comparison of test line 17 using LLMs for modes 1,2, and 7.



### Linearly-Fit LLM Parameter Results

Final analysis of the Test line 17 data is done by performing a linear fit of the LLM parameters to Mach number as described by Eq. 5-15. Rather than use the signal switch times determined by simplified GLRT, a linear Mach vector was used to fit the LLM parameters at a Mach resolution of 0.025M. The LLMs were used to construct a data sequence that is compared to the same flight test accelerometer desired response as before. This comparison is shown in Figure 5-17.

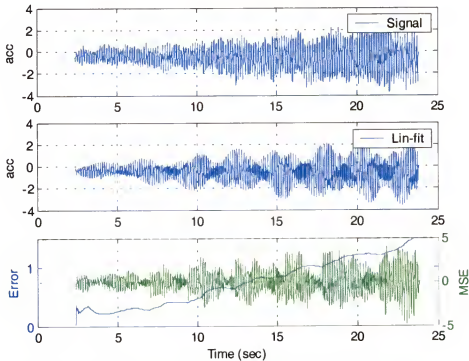


Figure 5-17 Data synthesis of test line 17 using linear-approximated LLMs.

The comparison is interesting because it shows both the ability to successfully model the important characteristics of the desired response, and the limitations of doing so. The envelope of the desired response is modeled very well, although influenced by an exaggeration of the beating tendencies in the signal. The envelope is the most important aspect of the approximation since it is from the envelope that safety of flight

for the aircraft configuration is determined. Throughout the sequence it is seen that any differences that exist between the desired response and the approximation are conservative. That is, the approximations tend to indicate normal accelerations that are slightly higher than those actually encountered, providing a margin of safety. Interesting to note is the high MSE and errors shown. Since the synthesized signal is a result of models changing at regular intervals rather than at switch times, the desired response and model outputs are out of phase. As noted, the envelopes match well, but value to value comparison is poor. The fact that the result is a few samples out of phase is of little consequence, however since the output is intended to simulate the response of the wing over a portion of flight.

The PSD for the linear-fit LLMs is shown in Figure 5-18. It is clear from a graphical comparison of the frequency contents of the desired response and its approximation that energy is distributed more evenly between the three active modes in the LLMs. All modes are represented, but the linear approximation of LLM parameters and the more even distribution of LLMs as a function of Mach number has effectively removed all but the three active modes of oscillation. The 7 Hz primary mode is now made up of the energy contributions of two close modes.

Results of Test line 17 LLM data synthesis, and analysis of the adapted LLM parameters show the potential of the LLM paradigm for use on-line for predicting wing responses based on previous and current model parameters. Further, it is seen that examination of LLM components can provide insight into the critical makeup of the modal behavior of LCO responses. These concepts are examined now by using the paradigm again to synthesis the Test line 3 g-loaded data.

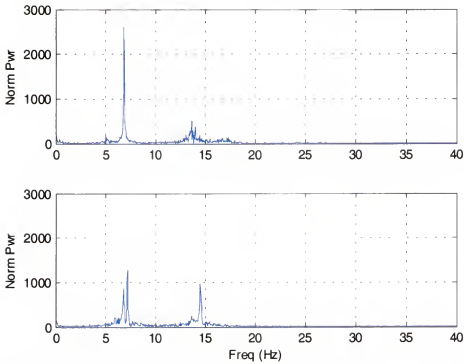


Figure 5-18 PSD comparison of data synthesis using linear-approximated LLMs.

### Test Line 3 LLM Experiments

The flight test conditions for Test line 3 consist of a step-wise progression from level flight to high g-loading of the aircraft. The Mach number for this test line was held constant, and again it is reasonable to claim that these tests are performed by changing only one independent variable, at least from a macro view. This is, of course, not truly the case since many factors are constantly changing throughout the flight, such as aircraft weight, air density, slight variations in airspeed and altitude, temperature changes, etc. But these are small compared to the intentional changes in normal acceleration made during testing. The difference between this test line and the previous is that the transition from one normal acceleration value to the next is not made in a regular, linearly progressive fashion. Instead the pilot transitions from one condition to the next in a maneuver referred to as a ‘wind up turn’ resulting in a somewhat complex combination of

aircraft role and changes in angle of attack, followed by brief periods of consistent flight at the desired test condition. Consequently the accelerometer signals for Test line 3 must be evaluated as discrete windows of data rather than as a continuous sequence.

Test line 3 is divided into five data sequences at nearly constant altitude and airspeed, with g-loadings of 1g, 2g, 3g, 4g, and 4.5g, at which point testing was terminated due to high LCO amplitudes. In order to model these sequences five LLMs were adapted at specific times rather than at times determined by the simplified GLRT. This will serve to illustrate the LLM paradigm's utility as a post-processing tool, providing insight into the construct and nature of the LCO response. Also, insight into the role played by normal acceleration on model parameters, and therefore on the LCO phenomenon itself can be examined.

Local linear models were adapted to each of the five data sequences. Since these were adapted prior to the discovery of network redundancy they were processed as described by equations 5-1 – 5-13 to a reduced-order set of FIR filters. Results of simulation with these adapted filters are plotted in Figure 5-19 (there was no difference between the second-order and first-order simulations). All five sequences are modeled very well. The 1g sequence is significantly longer than the sequences modeled during Mach testing. Since all parameters remained relatively constant for a longer period of time one LLM is sufficient to model the entire sequence. The 2g and 3g sequences are about one second in duration. The 4g sequence is only 0.8 seconds long, indicative of the difficulty of holding flight conditions constant, especially with increasing LCO. Finally the 4.5g sequence is only about 0.5 seconds. This is the terminating point of the test. Interesting to note is that the higher frequency modes in the signals tend to become less active as g-loading increases, perhaps due to wing stiffening.

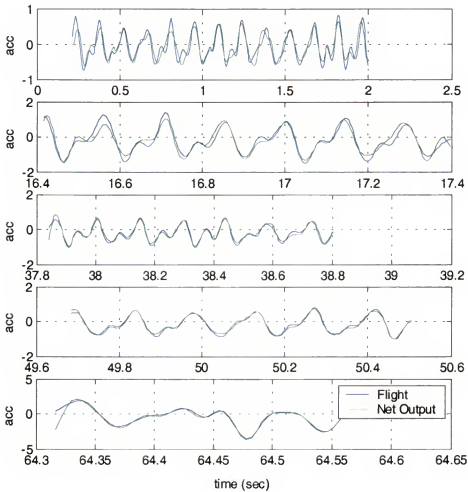


Figure 5-19 Test line 3 reduced order results.

### Third-Order Parametric Fit Prediction Results

The LLM parameters were all converted to weight vector magnitudes and angles as before. Figure 5-20 shows the weight vector angles, grouped by mode. There are five angular values per mode. The five for mode 1 are shown in the first set, followed by those for mode 2, etc. From this graph it can be seen that only two of the modes, the first and the seventh, behave in a linearly predictable manner. Most of the other modes are

difficult to describe in any orderly fashion, indicating that for the most part modes 1 and 7 are expected to be the primary modes of oscillation.

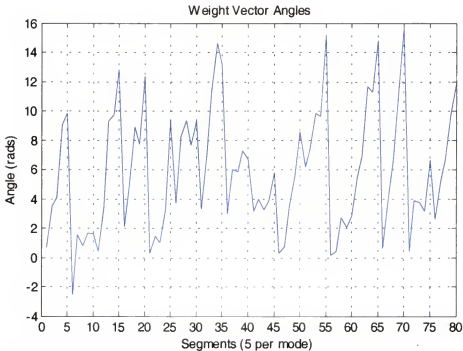


Figure 5-20 Test line 3 reduced-order LLM FIR modal vector angles by segments.

The relationship between the weight vector magnitudes  $\alpha$  and g-loading are plotted in Figure 5-21. There is no longer a linear relationship evident between them. Although there are only five data points for each mode from which to determine this relationship, it appears that a model for the vector magnitude as a function of normal acceleration must be at least third order. This result agrees, at least conceptually, with Cooper and Wright's<sup>29</sup> nonlinear dependency on velocity in their two degree-of-freedom work.

Magnitudes for most of the modes, especially those above mode 8, are very small except for the LLMs corresponding to near-divergent LCO at 4.5g. Magnitudes for modes 1, 7, and 8 seem the most significant and well behaved. Those of mode 2 oscillate around a small value, indicating that the influence of mode 2 on the simulated signal may

not be real. Further evidence for selecting these modes as active can be seen in the W1 vs. W2 plots in Figure 5-22. Equation 5-12 describes a linear relationship for these weights for a fixed frequency  $\omega$ . Active modes are once again indicated when this linear relationship is observed. Modes 1, 2, and 7 display this tendency.

As with the Test line 17 LLMs, looking at the behavior of polar plots of the weight vectors' magnitude and angle give a very good indication of the significance of the modes. These are plotted in Figure 5-23 for the Test line 3 LLMs. The plots are difficult to interpret since there are only five data points for each weight vector. But the results are consistent with the previous graphical observations. Modes 2 and 7 are the most symmetrically behaving modes of the set. The less significant modes do not display any discernable pattern in the polar plots.

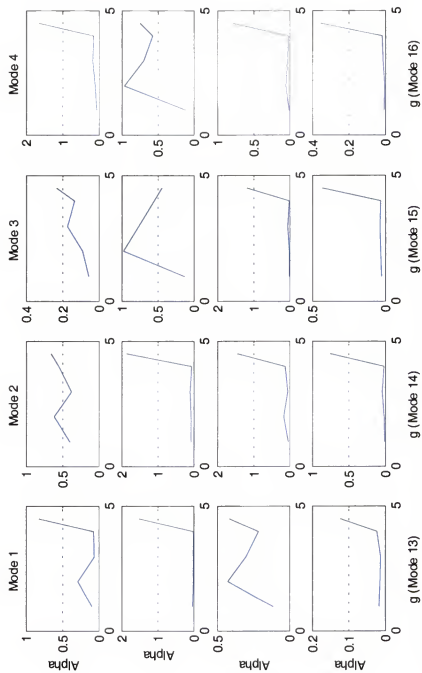


Figure 5-21 Test line 3 reduced-order FIR weight vector angles vs. g-loading.



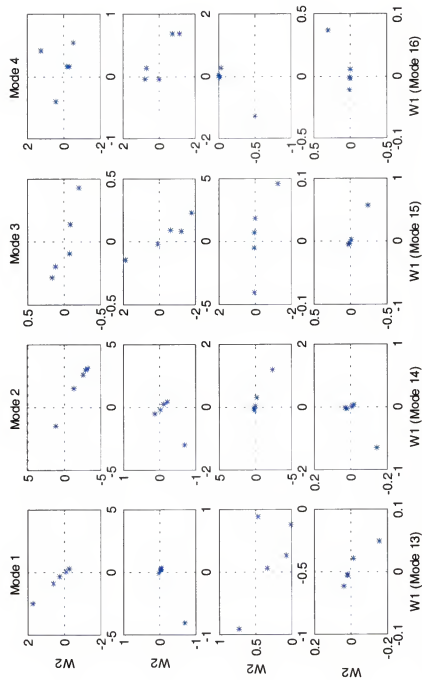


Figure 5-22 Test line 3 reduced-order FIR first-order vs. second-order weights.

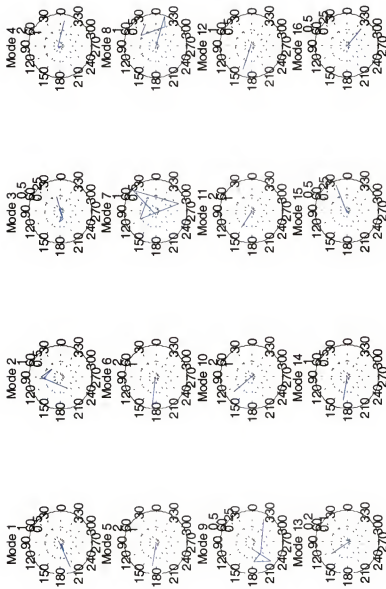


Figure 5-23 Test line 3 reduced-order FIR weight vectors.

### **Parametric Prediction Results**

In order to test the ability to predict LLM parameters a third order fit was made to the five LLMs as a function of normal acceleration. Simulations were made as before, and the results plotted against the desired response, shown in Figure 5-24. These approximations were not as successful as those of the linear Mach-fit parameters. The 1g data shows a good approximation. From there on, however, the ability of the approximations to match the data decreases. Phase and amplitude differences are noted in the 2g approximations, although the envelope of the predicted response is close to that of the desired response. The argument for acceptability can still be made from a macro point of view since LCO is predicted with magnitudes greater than or equal to the actual wing response, providing a margin of safety.

The 3g and 4g sequences are relatively good matches from the same point of view. The simulation is adequate to describe the envelope and thus prescribe safety of flight. Notice that there is an increase in signal magnitude towards the end of both sequences in the approximated outputs, showing that phases for the modes (a function of the FIR portions of the modal filters) may be inaccurate. The 4.5g approximations have large errors, but again towards the conservative side.

### **Reduced-Mode Results**

Finally, a test of the significant modes chosen is in order. Figure 5-25 shows simulations using LLMs containing modal information for modes 1, 2, and 7. Results are very similar to those of the third-order approximations. The 1g data is matched very well. The phase and frequencies of the 2g and 3g data sequences are also very good, but the magnitudes of the simulations are greater than those of the actual signal. The 4g simulation is quite good. It should be noted that the desired responses at 2g, 3g and 4g do

not have consistent magnitudes but instead tend to decrease towards the end of each sequence. The tri-modal approximations do not follow this trend, but instead provide a more consistent output. This is more clearly evident in the 4.5g data segment. The flight test signal, while short, is still quite dynamic. The tri-modal approximation is very consistent, however, and does not match the desired response well. It does provide very similar maximum and minimum amplitudes, though, and is therefore still useful for detecting safety of flight conditions.

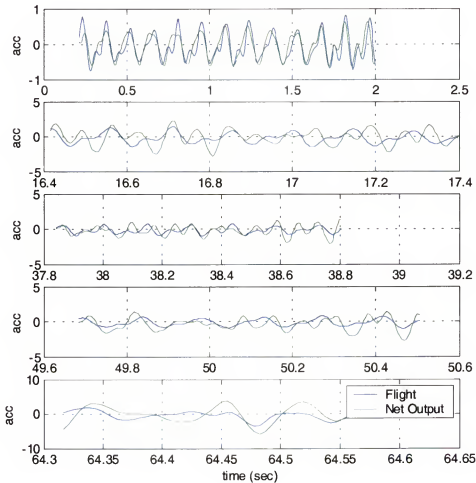


Figure 5-24 Data synthesis of Test line 3 using Third order approximated LLMs.

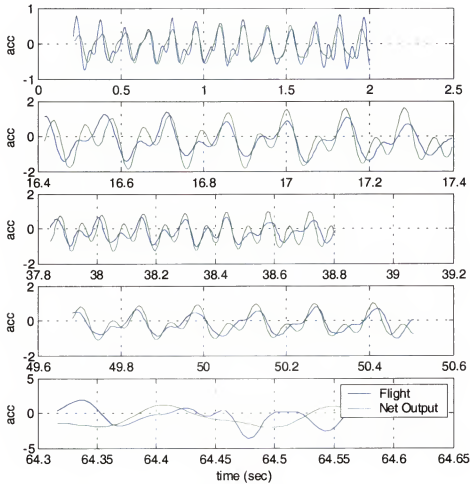


Figure 5-25 Data synthesis of Test line 17 using LLMs for modes 1,2, and 7.

### Frequency Adaptation Analysis

There is some expectation that the modal frequencies of the LCO oscillations may shift over the course of flight. The network architecture and its adapted parameters allow a unique evaluation of these frequencies. The frequency controlling weights of the linear oscillators provide a direct calculation for each mode. The results are encouraging, in that they show that while the relationship between LCO and flight test states may be nonlinear, the frequencies stay nearly constant throughout the flight. Table 5-1 shows the

frequencies for the Test line 17 data, modes 1 through 8. Modes 9 through 16 are shown in Table 5-2. Similarly, those for Test line 3 are shown in Table 5-3.

Table 5-1 Test line 17 modes 1 to 8 adapted frequencies (Hz)

Mode							
1	2	3	4	5	6	7	8
Free Vibration Frequency							
6.5460	7.0091	8.4156	8.5785	9.8033	12.3948	14.2342	14.6988
Adapted Frequencies							
6.5453	7.0085	8.4161	8.5780	9.8038	12.3951	14.2338	14.6985
6.5463	7.0079	8.4173	8.5775	9.8033	12.3936	14.2341	14.6985
6.5481	7.0072	8.4162	8.5780	9.8046	12.3907	14.2339	14.6987
6.5446	7.0064	8.4153	8.5786	9.8077	12.3893	14.2337	14.6988
6.5516	7.0079	8.4140	8.5796	9.8092	12.3880	14.2338	14.7002
6.5445	7.0098	8.4165	8.5798	9.8031	12.3975	14.2336	14.6986
6.5440	7.0078	8.4144	8.5807	9.8066	12.3883	14.2344	14.6990
6.5437	7.0098	8.4152	8.5761	9.8050	12.3933	14.2340	14.6986
6.5463	7.0089	8.4174	8.5793	9.8030	12.3952	14.2339	14.6977
6.5444	7.0087	8.4143	8.5784	9.8142	12.3928	14.2325	14.6990
6.5438	7.0096	8.4162	8.5782	9.8080	12.3882	14.2343	14.6986
6.5477	7.0062	8.4192	8.5764	9.8038	12.3904	14.2356	14.6991
6.5450	7.0087	8.4126	8.5757	9.8020	12.3895	14.2326	14.6996
6.5470	7.0080	8.4136	8.5767	9.8020	12.3929	14.2336	14.6981
6.5477	7.0064	8.4106	8.5722	9.7974	12.3897	14.2348	14.6987
6.5458	7.0076	8.4241	8.5823	9.8000	12.3910	14.2337	14.6988
6.5449	7.0084	8.4202	8.5825	9.8020	12.3948	14.2339	14.6981
6.5451	7.0088	8.4216	8.5825	9.8007	12.3932	14.2338	14.6984
6.5466	7.0078	8.4145	8.5774	9.7997	12.3928	14.2336	14.6976
6.5441	7.0093	8.4146	8.5766	9.8039	12.3925	14.2336	14.6985
6.5443	7.0092	8.4185	8.5755	9.8089	12.3897	14.2338	14.6989
6.5459	7.0084	8.4162	8.5795	9.8094	12.3905	14.2343	14.7000
6.5441	7.0099	8.4154	8.5772	9.8101	12.3934	14.2338	14.6987
6.5407	7.0102	8.4211	8.5809	9.8127	12.3892	14.2352	14.7004
6.5407	7.0102	8.4211	8.5809	9.8127	12.3892	14.2352	14.7004

The frequencies for the active modes vary only slightly from LLM to LLM, an indication that for practical purposes the modal frequencies are remaining consistent over the course of the flight, regardless of changing dynamics that are encountered. This makes intuitive sense because frequency is heavily influenced by the structural characteristics of the wing; mass and stiffness.

Table 5-2 Test line 17 modes 9 to 16 adapted frequencies (Hz)

Mode							
9	10	11	12	13	14	15	16
Free Vibration Frequency							
15.2114	16.4557	17.8903	19.3882	22.9792	26.1874	26.9629	30.2262
Adapted Frequencies							
15.2111	16.4560	17.8900	19.3884	22.9794	26.1876	26.9630	30.2264
15.2110	16.4550	17.8912	19.3886	22.9799	26.1866	26.9624	30.2255
15.2112	16.4552	17.8921	19.3876	22.9804	26.1867	26.9614	30.2251
15.2122	16.4564	17.8919	19.3904	22.9802	26.1886	26.9615	30.2247
15.2119	16.4556	17.8917	19.3894	22.9798	26.1879	26.9623	30.2257
15.2097	16.4548	17.8913	19.3879	22.9799	26.1864	26.9639	30.2258
15.2127	16.4565	17.8913	19.3914	22.9798	26.1901	26.9628	30.2243
15.2105	16.4565	17.8907	19.3891	22.9802	26.1877	26.9613	30.2268
15.2114	16.4566	17.8916	19.3892	22.9792	26.1862	26.9628	30.2256
15.2123	16.4567	17.8926	19.3838	22.9810	26.1912	26.9627	30.2258
15.2131	16.4570	17.8914	19.3909	22.9786	26.1896	26.9626	30.2238
15.2122	16.4573	17.8852	19.3900	22.9810	26.1836	26.9649	30.2299
15.2093	16.4549	17.8913	19.3938	22.9780	26.1901	26.9642	30.2216
15.2102	16.4529	17.8913	19.3909	22.9798	26.1906	26.9670	30.2216
15.2096	16.4533	17.8914	19.3926	22.9798	26.1910	26.9687	30.2226
15.2084	16.4547	17.8926	19.3893	22.9815	26.1883	26.9566	30.2292
15.2105	16.4569	17.8921	19.3833	22.9832	26.1910	26.9592	30.2296
15.2101	16.4573	17.8920	19.3819	22.9819	26.1904	26.9598	30.2296
15.2088	16.4533	17.8912	19.3914	22.9785	26.1901	26.9679	30.2208
15.2101	16.4552	17.8903	19.3910	22.9778	26.1931	26.9672	30.2204
15.2120	16.4572	17.8957	19.3921	22.9796	26.1943	26.9627	30.2260
15.2126	16.4567	17.8929	19.3893	22.9774	26.1900	26.9651	30.2252
15.2111	16.4554	17.8920	19.3913	22.9814	26.1848	26.9614	30.2271
15.2143	16.4601	17.8995	19.3924	22.9770	26.1891	26.9609	30.2294
15.2143	16.4601	17.8995	19.3924	22.9770	26.1891	26.9609	30.2294

Results show that most of the frequency changes are made in the third and forth decimal place. Since the frequency resolution for most of the sequences is about 0.009 Hz it can be assumed that of these changes don't necessarily reflect any actual frequency change but simply follow the gradient slope of the error during backpropagation. Also, there is some oscillatory variation of the frequency values of the higher frequencies. These are of much lesser importance, their magnitudes are for the most part negated with near-zero FIR filter attenuation and can be attributed to noise. The adapted frequencies for the more active modes are more consistent.

Table 5-3 Test line 3 adapted frequencies (Hz)

Mode							
1	2	3	4	5	6	7	8
Free Vibration Frequency							
6.5460	7.0091	8.4156	8.5785	9.8033	12.3948	14.2342	14.6988
Adapted Frequencies							
6.5773	7.0012	8.3904	8.5920	9.8167	12.4014	14.2429	14.6977
6.5767	7.0038	8.3891	8.5834	9.8074	12.3965	14.2397	14.6928
6.5490	7.0029	8.4122	8.5833	9.8042	12.3983	14.2420	14.6948
6.5620	7.0029	8.4050	8.5799	9.8032	12.3988	14.2420	14.6951
6.5617	7.0024	8.4063	8.5806	9.8043	12.3994	14.2410	14.6957
Mode							
9	10	11	12	13	14	15	16
Free Vibration Frequency							
15.2114	16.4557	17.8903	19.3882	22.9792	26.1874	26.9629	30.2262
Adapted Frequencies							
15.2200	16.4598	17.8901	19.3875	22.9945	26.1851	26.9600	30.2278
15.2144	16.4583	17.8928	19.3911	22.9891	26.1863	26.9620	30.2257
15.2142	16.4581	17.8934	19.3892	22.9846	26.1852	26.9612	30.2250
15.2155	16.4586	17.8915	19.3880	22.9871	26.1849	26.9607	30.2265
15.2150	16.4585	17.8916	19.3884	22.9856	26.1849	26.9606	30.2266

### Model Generalization Evaluation

As an exercise to evaluate the generalization of the LLMs, the first model for Test line 17, associated with a normal acceleration of 1g and Mach number of 0.78, flown at 5,000 feet, is applied to the first segment of data from Test line 3. This test was also at 1g, with a Mach number of 0.8, and at an altitude of 10,000 feet. The comparison is shown in Figure 5-26. The LLM from Test line 17 does not fit the data exactly, but does provide a close enough approximation to give very reasonable estimates of the LCO that is encountered. Most important to note is that the bias is not necessary. The bias noticed during Test line 17 is not present in this data set. Therefore it must be ignored when applying the model.

There are two critical differences between the conditions under which these tests were conducted that may help to explain this. While the Mach numbers are quite close, the calibrated airspeeds for the two are not the same because of the altitude differences.



Referring again to the KCAS, Mach, altitude chart provided in Chapter 2, Figure 2-5, the calibrated airspeed value associated with the Test line 17 (0.78M) model is about 453 KCAS, while that of Test line 3 segment 1 (0.8M) is 450 KCAS. Intuitively it would be expected that the second LLM from Test line 17 would be the better match since it's Mach number is 0.8. This is not the case, however, since the calibrated airspeeds of Test line 17 LLM 1 is closest to that of the Test line 3 segment 1.

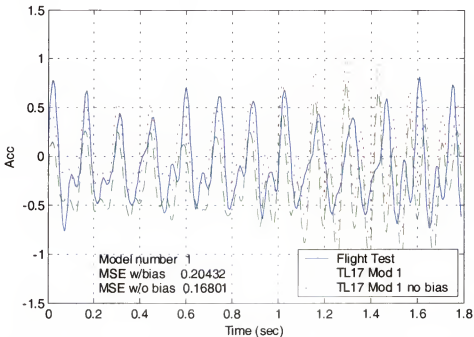


Figure 5-26 Data synthesis of test line 3 segment 1, LLM for Test line 17 Model 1.

The second critical difference between data sets lies in the manner of testing. Test line 17 is a test flown in level flight with a consistently increasing airspeed. It can be argued that other variables of flight will be changing as well. Most significant is angle of attack. As airspeed increases with all other variables remaining constant, angle of attack decreases in a similar, linear fashion. Thus a bias in the direction of the LCO responses can be expected. Contrast this to the Test line 3 conditions. The goal of that test was to

maintain constant normal acceleration, airspeed and altitude for a period of time. It is expected, then, that angle of attack will also remain constant and the bias in LCO would not be as noticeable, certainly not comparable to that of Test line 17.

The best fit occurs during the first second of the data. There appears to be a phase shift in the desired response at about 1.25 seconds. The linear model is not able to adjust to follow. The envelope of the desired response is well represented by the LLM and as with prior models, would provide an accurate yet conservative estimate of the expected LCO response. The overall MSE for the model is 0.204 for the LLM when the bias is included in the LLM, and 0.168 without applying the bias.

### **Chapter Summary**

Experiments evaluating the use of LLMs to synthesize flight test accelerometer data were evaluated in this chapter. The paradigm designed in Chapters 3 and 4 was implemented and results presented for two flight test sequences. The first was the data from Test line 17, a signal showing increasing LCO with increasing Mach number. The second was a series of five data sequences taken from Test line 3 where the varying independent variable was normal acceleration. Test line 17 data was chosen to evaluate the paradigm's use as an on-line data synthesis tool. Test line 3 data was chosen to evaluate it's use as a post-processing analysis method.

The interesting results from the Test line 17 experiments were that the signal was successfully modeled with a total of 25 sequential LLMs each adapted to fit data sequences of about one second. A bias towards negative acceleration was observed and successfully modeled by the LLMs. Each model used parameters from the previous model and adaptation occurred within 150 epochs.

There was a redundancy noted in the FIR model parameters and a method was presented for reducing the FIR filter order from second to first. This was successfully accomplished. Data synthesis was done using the reduced-order models with no degradation to the signal. During analysis of the FIR filter order it was noted that the model parameters could themselves be modeled as functions of Mach number and modal frequency. This was accomplished and results provided. Finally, experimental results showed that there were at most three active modes in the signal. The LLMs for those modes were selected and used in a reduced-mode LLM with good results.

Next the test data from Test line 3 was evaluated using the paradigm. The order of the adapted LLMs was reduced from second-order to first-order, again with no degradation to the data synthesis. The model parameters were evaluated and an attempt was made to model them using a third-order fit as a function of normal acceleration. It was shown that although this remains a promising possibility, the output from the fit-models was not as good as the adapted models. Again the more active modes were identified from model parameters and a reduced-mode simulation was performed. Results were again promising but not as accurate as directly adapted models.

Parametric analysis of both data sets showed that the proposed paradigm has a high potential to identify active modes in an oscillatory response.

Finally, an evaluation was made intended to demonstrate the ability of a LLM adapted with one set of data to generalize to a similar but unknown data set. The first adapted LLM for the Test line 17 data was applied to the first data sequence from Test line 3. These two tests had similar flight conditions, but differed in calibrated airspeed and altitude. The results showed that the model's output, while not an exact match, was fully sufficient to predict the trend in an LCO response from which a safety of flight

judgement could be made. Errors with this experiment were on the conservative side, as they were with all experimental results in this chapter.

## CHAPTER 6 DISCUSSION AND CONCLUSIONS

This chapter addresses the results, conclusions and implications of the experiments conducted in this work in an effort to develop, define, and refine a paradigm for use for modeling flight test accelerometer data using local linear models. Several conclusions can be drawn and are addressed. As with any research project there are questions left unanswered; open topics for further research and investigation. The chapter is organized to present the overall conclusion and conclusions associated with modeling and model components followed by a discussion of topics for further analysis and research. The chapter concludes by briefly evaluating a confirmation of existing linear flutter methods resulting from analysis of model components in this work.

Overall it can be stated that the hybrid physical model to synthesize the accelerometer data from the wing tip on-board instrumentation was a success, at least for level flight. The experimental results in chapter 5, experiments using the paradigm and model architecture to synthesize two entirely separate portions of flight under distinctly differing test conditions, show that a sequence of component-based local linear models consisting of quasi-fixed IIR pole-placing subnetworks driving a set of adaptive FIR filters can be used to synthesize the IIR LCO response of the wing under dynamic aeroelastic forces, and can provide descriptions of all modes contained in the model, effectively decomposing the flight test signal into its modal representation.

The usefulness of this paradigm as an on-line tool was clearly demonstrated. Noting that the signal from Test line 17 was very long reinforces this conclusion. The

system was able to continuously model a changing wing response without fail for 25 seconds. The results of experiments using that data, exploiting the concept of building a continuous series of sequential LLMs to model a signal, can easily be interpreted to imply that the system could be used to model signals for an indefinite period of time, provided that the hardware used can keep up with the calculations required.

The computational burden is eased by a number of factors. The LLMs, once initialized, are used as initial conditions for each LLM in the sequence. This means that the adaptation process does not have to search the entire solution space every time a model is trained. It simply has to fine-tune the previous model. It was further demonstrated that in many cases a fixed interval between LLMs was sufficient. This may be used instead of the simplified GLRT when flight dynamics are steady, keeping the number of required computations to a minimum. When networks are originally initialized, or when dynamics rapidly and severely change, it was shown that linear flutter analysis results provide a good starting point from which to search for a final solution.

The paradigm performed well as a post-processing tool as was demonstrated with the Test line 3 experiments. The data simulations were as good as those of the Test line 17 on-line experiments, but the dynamics were much more difficult to define in terms of flight test states. It was shown that overall the model parameters could be described as functions of normal acceleration. The Test line 17 model parameters were successfully modeled as functions of Mach number. The fact that model parameters can be modeled as functions of test state opens new avenues of research. It has been assumed that LCO could be modeled as nonlinear functions of states, but there has been little test data to base any firm modeling conclusions on since flight test data is simply a measure of a total response. The LLM paradigm contained in this work now gives insight into the modal

components that make up these signals. Since parameters of these models can now be modeled as functions of test states, it follows that a more definitive form of model can be constructed for the LCO phenomenon based on those parametric models. This is perhaps the most useful aspect of this work as it applies to practical, day-to-day LCO and flutter testing and analysis.

It was shown that the LLMs are useful for generalizing over a wider solution space, provided there is some a priori knowledge of the system applied. This was demonstrated by using a model from the Test line 17 data to simulate the response of the first segment of the Test line 3 data. It was shown that this was a viable option for using the models for similar but separate flight conditions, provided those differences were understood. In this case the differences are in altitude and airspeed, as well as subtle changes in the way the test was performed. Specifically, the angle of attack changes in the Test line 17 data caused a bias to be required to model the data correctly. That bias was not noted in the Test line 3 responses, however. As long as it is understood that these differences can be expected and explained, using models across a wider portion of flight is reasonable. This further eases the computational burden when performing on-line signal analysis.

### **Modeling Conclusions**

Now that the paradigm has been shown to be a success, conclusions based on different elements of the paradigm and its components can be addressed.

#### **Linear Oscillator**

It was shown that a small, adaptive, two-PE linear recursive network could correctly place the poles in the IIR modular components, providing the sinusoidal inputs for the FIR filters. This network was derived from classical spring-mass differential equations. The degrees of freedom in the network were reduced to a single adaptable

weight, allowing the frequency of oscillation to be tuned to a desired value, provided the desired frequency is within acceptable Nyquist limits; and the frequency controlling weight is initialized to a value resulting in an initial frequency in the neighborhood of the desired value. The oscillator is driven by a single impulse to each of its two inputs, and will oscillate indefinitely thereafter with no further input.

For this work the oscillators were set to provide true sine waves, that is, oscillations beginning at zero. The relative phases between the frequencies in the signals were adjusted by the FIR filters as part of the larger architecture. It is possible to adjust the phase of the oscillators within the oscillator networks by setting the input weights to appropriate values based on the frequency. It was shown that the preferred method for setting these weights was to adapt the frequency controlling weight using some form of BPTT, and adjusting the input weights based on the results after each epoch.

These adaptive oscillators are an important component of the LLM, since they allow the poles of the system to be independently set and adjusted as required, separate from each other and the zeros of the system.

### **Fixed-length Signal Modeling**

The fixed-length signal modeling experiments in Chapter 4 focused on evaluating the practical limitations of the linear models to model the nonlinear LCO responses. For all test cases analyzed it was shown that trained networks could model accelerometer signals for at least one second. This conclusion is valid for signals under steady-state conditions as well as for conditions that are linearly changing during flight, as demonstrated by using sequential networks trained at fixed intervals to model a portion of the Test line 17 signal. This was further validated in later tests using a simplified GLRT to detect signal changes. The system paradigm used in Chapter 4, including the



simplified GLRT, showed that parametric adjustment was required at intervals just over one second for the same data set.

An interesting side note pertaining to the conclusion of a one-second signal length is that the Test line 3 studies also used signals whose lengths averaged about one second. These signals were manually selected to represent responses during times of constant flight. It turns out that the test pilot was only able to maintain steady conditions for short periods of time, usually more or less one second, when conditions were dynamic and in the presence of significant LCO. This further reinforces the notion that LLMs valid for one second of flight time are very adequate for LCO analysis.

### **Reduced-Model Simplified GLRT**

It is concluded that reduced-model simplified GLRT is an acceptable algorithm to detect changes in the desired response sufficient to require network adjustment. Further, the statistics contained in simplified GLRT can be used to indicate the nearest set of network parameters to use as initial conditions for retraining, if trained networks exist in its database. This was demonstrated not only by the extensive experimental build-up in Chapter 4, which sought to mimic an expected bimodal LCO in the experimental desired response, but by the Test line 17 experiments as well. It was shown that the simplified GLRT proved sufficient to identify signal change-times as a function of normally distributed statistical evaluation of the signal. The simplified GLRT was not necessarily shown to be the most efficient method for detecting changes, however. It is admittedly quite computationally intensive, requiring the statistics of all available models to be continuously updated and compared. Anecdotal evaluation of execution times showed that many times signal detection took quite a bit longer to accomplish than retraining the model once a change was identified.

There may be other methods by which signal changes can be detected. As an alternative it may be advantageous in a practical implementation of the paradigm to simply retrain or fine-tune the models at regular or semi-regular intervals based on changes to flight conditions rather than, or in addition to a simpler statistical evaluation of the LLM currently in use. These possibilities were not explored since there are obvious limitations that may result. Segmentation based solely on flight test state will in all likelihood result in missed dynamics. It is clear that for a given test state the response of the wing may not be at a steady state condition. In all likelihood several LLMs would be required to synthesize a response from onset to steady behavior. Clearly, whatever method is chosen to segment the data must take this into account.

### **Flight Test Data Experiments**

The conclusion that a flight test signal, a time varying IIR response, whose conditions are changing at a controlled rate can be successfully synthesized by a sequence of component-based local linear models consisting of quasi-fixed IIR pole-placing subnetworks driving a set of adaptive FIR filters has been stated. This was demonstrated by the success of the experiments using the Test line 17 and Test line 3 data. Networks can be retrained as changes in the signal warrant, or simply by retraining at regular intervals. There are several modeling conclusions that can be drawn as consequences to this.

The paradigm provides several indicators that identify the active modes in the accelerometer signals. As a test of the accuracy of these, signals were synthesized using only LLMs corresponding to those modes identified as active. For both Test lines there were three modes identified as active. It was interesting to note that in fact it was the same three modes identified in both test sets. Models for these were used, and it was

shown that the signals could be synthesized very well. The tri-modal LLMs resulted in a signal simulation that could be used to determine safe flight conditions since in all cases the error between modeled and actual responses was conservative. These determinations are subjective, but data synthesis was deemed successful if the question of whether or not the synthesized signal could be used by engineers observing its behavior to determine approximate LCO conditions and safety of flight.

One conclusion drawn from the Reduced-Model simplified GLRT experiments in Chapter 4 was that of the two methods for modeling, retraining or tuning versus switching between known models, the option to switch between models was preferred. However, the flight test experiments in Chapter 5 used model training rather than switching. There were two reasons for this seemingly diversion from the recommended methodology.

Primarily it was concluded that the best way to conduct the initial flight test experiments was as if they were the first application of the paradigm. As such there would be no models from which to choose, other than that of the beginning of the signal. These experiments were conducted and shown to be successful. Subsequent to those experiments, the Test 17 data set was segmented by hand into 12 segments and models trained to the first second of each. These models were stored in a database and the segmentation and selection paradigm again run. The experimental results are not contained in this work per se since they are virtually identical to those of the original experiments. The noteworthy conclusion was that the segmentation algorithm segmented the data at the same points as in the prior experiments as expected, and, since in most instances no model in the database matched at that time, a new model was trained rather than selecting from those selected by hand. This is further evidence of the ability of the

paradigm to accurately determine when model training is necessary, and also serves to illustrate that it is difficult and usually incorrect to segment data strictly by casual observation when the dynamics of the system are continually changing as they are in the Test line 17 data set.

There are times, however, when segmentation by hand is the only reasonable option, as was the case with the Test line 3 data. As noted, the dynamics of that portion of flight left little choice but to segment the data at portions of flight where all conditions were consistent. It cannot be concluded that a single model was sufficient for modeling a single flight condition from those experimental results, however. It turned out that for each of the five data windows a single model was all that was required, but that certainly cannot be extrapolated to conclude that would be the case under all conditions. It is coincidental and a fact of the nature of flight test that the duration of the responses under those conditions was short. It is easy to envision circumstances where this would not be the case and even though segmentation by hand is necessary to start the process, further automatic segmentation could be necessary.

It was shown that for portions of flight whose conditions are changing on Mach number the network parameters could be modeled as linear functions of Mach. The ability to predict model parameters as a function of flight test state provides a capability to the network for extrapolating wing response beyond the current conditions. This was demonstrated during the Test line 17 experiments with very good success. The finer details of the signal were not present, but the overall envelope of the signal as well represented throughout the entire 25-second test duration. Again, the criteria used for evaluation was that of a conservative estimate of the signal envelope.

For flight tests where normal acceleration is the changing variable, such as those in Test line 3, parametric modeling became more difficult. Flight conditions between test points are often very chaotic until flight is stabilized at the desired test condition. Consequently it is of little use to attempt on-line modeling of interim flight conditions. Networks were successfully trained at the steady-state g-loaded conditions, however. This shows that the synthesis algorithm is very useful as a post-processing tool, providing insight into modal behavior as a function of normal acceleration.

Parametric evaluation of the networks trained with the five Test line 3 points showed a higher-order relationship between model parameters and normal acceleration. A third-order polynomial fit was used with some success. But since there was a limited amount of data at steady state conditions for this analysis, it is difficult to draw a firm conclusion regarding the ability of the network parameters to be predicted as a function of normal acceleration.

Two conclusions can be drawn from parametric analysis of the Test line 3 networks, however. First, that as loading increases the higher frequency modes are more strongly damped. This makes sense from a structural standpoint because of the biased bending force exerted on the wing, in effect stiffening the wing.

Second, that the modal phase and gains are not linear functions of normal acceleration. The modal gains for active modes consistently appear to be at least third-order functions. The modal phases no longer appear linear for many of the modes, although they are linear for the most active modes. This leads to a likely conclusion that network parameters for active modes can be predicted and used as was demonstrated in the Test line 17 experiments, but there is insufficient data for a clear determination of the relationship of parameters to test state, nor a definitive demonstration in this work.

It was shown that the FIR network architecture with more than one delay was redundant, and that an equivalent single-delay network could be derived from the trained weights. A method by which this conversion can be accomplished was demonstrated. It is intuitive that two FIR weights would be sufficient to adjust the phase and gain of each mode. It is concluded that a single-delay FIR structure is sufficient.

Finally, it was shown that the IIR system being modeled, i.e., the wing's nonlinear LCO response, was successfully modeled with a LLM process that can be considered FIR in a global sense. That is, given the clustered or fitted network parameters as a function of aircraft state, with aircraft state as the input to the system, the output is entirely dependent on that input with no feedback connection from output to input. From that point of view the overall system is feedforward and it can be concluded that an IIR signal response has been successfully represented with an FIR system.

### **Further analyses**

Several topics for further research and experimentation are suggested from the results presented in this work. The experiments conducted showed the viability for the LLM paradigm for data synthesis, as well as opening new points of view on the construct of the LCO phenomenon. It follows that further evaluation of the use of parametric modeling and approximation may help to answer some of the questions that deal with the signals and their characteristics.

It was shown that in one set of test conditions the LLMs could be approximated very well as linear functions of Mach number, while in others higher order functions were required on normal acceleration. There was a common model between sets, however, that suggested perhaps a relationship on KCAS might have been a more desirable independent variable upon which to base a model than Mach. The results

indicate that a more obvious generalization or clustering of models over a larger solution space would have been observed. Consider again the results of using the first LLM from Test line 17 as it was applied to the first Test line 3 sequence. Intuitively it was thought the models shouldn't match because their Mach numbers differed. The second or third Test line 17 LLM had the proper Mach number, but was not the best fit on the Test line 3 sequence. A closer examination of the two data sets revealed that the two models used were in fact very close when the airspeeds were compared in units of KCAS. This is explained by the altitude difference between the two test conditions. This example serves to make the point that there may be different combinations of test states that can be used to cluster models in a more general manner.

This can be expanded to include higher order functions of flight test states for modeling or clustering LLM parameters. Consider again the Test line 17 data, modeled as a linear function of Mach. It was noted that the model biases appeared linear on Mach during most of the sequence, but that the larger residual errors could be expected as Mach number increased. Perhaps a model based on KCAS or a higher order on Mach, or a combination of Mach number and altitude would have served to better represent or 'store' the parameters. Similarly the third-order fit on normal acceleration from Test line 3 data certainly was less than optimal. The results did serve to illustrate that the LCO phenomenon must be modeled on nonlinear combinations of independent variables, but that the method chosen was most likely not the best that could be found.

As was noted in the discussion of the flight test data that the modal behavior of the signal in the 14Hz band may be the result of an harmonic of the primary 7Hz flutter mode. This was initially suspected because the 14Hz component was evident at the launcher tip accelerometer but nearly nonexistent at the launcher tail. It was speculated

that the higher frequency was a local aeroelastic phenomenon at the forward end of the launcher since it is a long, narrow protrusion. Determining whether or not this is the case is difficult and was not intended as a part of this work. However there was some anecdotal evidence provided that supports the conclusion that the behavior is indeed due to an harmonic. The gains of the filters associated with those frequencies were significant, but oscillated quite a bit, possibly indicating that the higher frequency is an harmonic of the lower. The data used in this test is somewhat atypical of the usual responses noted during flight in that more than one flutter mode seemed present. It is unusual for higher frequencies to be noted. This may well be an instance of an harmonic, but it would be interesting to do further investigation to determine this for certain. Possible approaches might be to use a coherence analysis on the data itself, or on the changing network parameters. Another alternative would be to make use of nonlinear networks, an interesting modification problem to the current design in that the challenge would be to incorporate nonlinear PEs in a network while at the same time maintaining the ability to interpret the physical meaning of training results as can be done with the linear networks.

Another interesting area of research that can exploit the results of this work would use of parametric models to further improve the current empirical modeling methods. At the present time linear flutter analysis considers all forces on the wing to be linear combinations of structural or aerodynamic influences. Consider the information already obtained from the experiments in this work. The linear relationship of model parameters to Mach tends to reinforce the original assumptions. However, the fact that a third-order approximation on the second time-derivative of wing deflection (normal acceleration) provides a viable simulation illustrates a limitation of current methods. The implication



is that by modeling more data, under a wider variety of flight conditions and then analyzing the relationships between the adapted LLMs as functions of independent variables can provide insight into the nature of the forces involved, perhaps sufficiently to further define the current empirical models.

Automatic clustering algorithms can also be used to determine the inter-relationships of test states and LLM parameters. By providing a clustering algorithm with sufficient state-space into which to map model parameters, the clustering parameters can then be evaluated for empirical relationships. Clustering wasn't practical for this work since only two sets of data with two independent variables were evaluated, with at most one condition that could be considered common. The purpose of the inter-model evaluation at the end of Chapter 5 was specifically included to address the issue of model generalization and clustering. Further research in this area would most likely provide a much clearer answer as to the more proper forms of empirical models that can be used to approximate the LCO and flutter phenomenon.

The most practical use for this work is as an on-line model change detection and prediction scheme for flight safety determination. Currently a flight must be monitored at all times by engineers watching for tell-tale signs of the onset of unsafe flight conditions. By using the paradigms described a prediction method can be set up providing the expected response ahead of the flight conditions, either to provide a warning or to ensure continued safe testing. It was shown that under carefully controlled conditions this is possible and viable. Since all testing must be done under extremely controlled conditions the utility of this paradigm is immediate. All tests are flown with data telemetry equipment that allows real-time data processing on the ground, at high speed, during the flight. The methods presented here can be implemented and used with the technology

currently available. Fail-safes can be built in to insure against continued flight in unsafe conditions when the predictions are uncertain, when data transmission fails, etc.

### **Confirmation of existing methods**

This work relied a great deal upon current practices for the prediction and evaluation of LCO and flutter in thin wing aircraft. Linear flutter analysis as described in Chapter 2 is still the backbone of the industry. It is simply too expensive to use other methods for practical flutter analysis at this time. Since traditional methods are so heavily depended upon, it is good to be able to provide insight into their accuracy whenever the opportunity presents itself, as is the case with this work. The fundamental basis for linear flutter analysis is that the solutions of its equations can be solved by assuming that flutter, and by extension LCO, can be considered a phenomenon that can be locally modeled as a sum of sinusoids. Based on the success of the modularly designed LLMs to synthesize the flight test data, those LLMs themselves representing a sum of sinusoids, it is reasonable to conclude that the assumption made by linear flutter methods that flutter can be modeled as such is correct. It was shown that this assumption holds for steady-state LCO responses at least for fixed state variables, i.e., altitude, Mach number, normal acceleration and aircraft configuration.

Further, linear flutter analysis provides modal frequencies that are in close proximity of the desired values. Based on the analysis of the modal content of both sets of data used in Chapter 5, it can be concluded that linear flutter theory's approximation of these frequencies are valid. It was shown that when the linear oscillator networks were initialized to these values very little adjustment was required. Since these frequencies are themselves a result of the linear eigen analysis performed during linear flutter analysis,

this result also serves to validate that the basic underlying structural assumptions of linear flutter analysis are correct.

## LIST OF REFERENCES

1. C. M. Denegri, Jr., "Limit Cycle Oscillation Flight Test Results of a Fighter with External Stores", Journal of Aircraft, Volume 37, Number 5, pp. 761-769.
2. C. M. Denegri, Jr., J. A. Dubben, "In-Flight Wing Deformation Characteristics During Limit Cycle Oscillations," 44<sup>th</sup> AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference, 7-10 April, 2003, Norfolk, Virginia, AIAA Paper 2003-1426.
3. A. M. Cunningham, Jr., "Semi-Empirical Unsteady Aerodynamics for Modeling Aircraft Limit Cycle Oscillations and Other Non-Linear Aeroelastic Problems," Lockheed Corporation, Fort Worth Texas, 1995.
4. C. M. Degegri, Jr., M. A. Cutchins, "Evaluation of Classical Flutter Analyses for the Prediction of Limit Cycle Oscillations," 38<sup>th</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 3 April 1997.
5. A. S. Weigend, N. A. Gershenfeld (Editors), *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, Boston, Massachusetts, 1994.
6. K. S. Narendra and K. Parthasarathy, *Identification and Control of Dynamical Systems Using Neural Networks*, IEEE Transactions on Neural Networks, Vol. 1, March 1990.
7. C. M. Denegri, Jr., M. R. Johnson, "Limit Cycle Oscillation Prediction Using Artificial Neural Networks", Journal of Guidance, Control, and Dynamics, Vol. 24, No. 5, pp. 887-895.
8. M. R. Johnson, C. M. Denegri, Jr., "Comparison of Static and Dynamic Neural Networks For Limit Cycle oscillation Prediction," Journal of Aircraft, Volume 40, Number 1, pp. 194-203.
9. M. R. Johnson, C. M. Denegri, Jr., "Statistical Data Evaluation For Limit Cycle Oscillation Prediction Using Artificial Neural Networks," 41<sup>st</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Atlanta, Georgia, 3 Apr 2000.
10. J. C. Principe, L. Wang, M. Motter, "Local dynamic modeling with self-organizing maps and applications to nonlinear system identification and control," Proc. of IEEE, Vol. 86, No. 11, pp. 2240-2258, Nov. 1998.

11. M. Motter, *'Control of the NASA Transonic Wind Tunnel with the Self-Organizing Feature Map,'* Ph.D. dissertation, University of Florida, Gainesville, Florida, Dec. 1997.
12. Sandberg, Lo, Fancourt, Principe, Katagiri, Haykin, *'Nonlinear Dynamical Systems,'* John Wiley and Sons, Inc., New York, New York, 2001.
13. C. M. Denegri, Jr., *'Characterization of Structural and Aerodynamic Stiffness and Damping Associated with Wing Limit Cycle Oscillations,'* Ph.D. Dissertation, Auburn University, Auburn, Alabama, June 12, 1998.
14. W. P. Rodden, *'Static Aeroelasticity and Flutter,'* Course notes and compendium, September 28-October 2, 1992, University of Kansas, Overland Park, Kansas.
15. J. P. Den Hartog, *'Mechanical Vibrations,'* 4<sup>th</sup> Edition, McGraw-Hill, New York, New York, 1956.
16. T. D. Burton, *Introduction to Dynamic Systems Analysis,* "McGraw-Hill, New York, New York, 1994, pp. 342-424.
17. E. Albano, W. P. Rodden, *'A Doublet-Lattice Method for Calculating Lift Distributions on Oscillating Surfaces in Subsonic Flows,'* AIAA Journal, Volume 7, Number 2, February 1969, pp. 279-285.
18. R. N. Desmarais, R. M. Bennett, *'An Automated Procedure for Computing Flutter Eigenvalues,'* Journal of Aircraft, Vol. 11, No. 2, 1974, pp. 75-80.
19. A. Albert, *'Regression and the Moore-Penrose Pseudoinverse,'* Academic Press, New York and London, 1972.
20. W. W. Dyess, Jr., D. Stokes, *'The Role of Digital Models and Simulations in the Air Force SEEK EAGLE Office,'* Symposium on Military, Government and Aerospace Simulation for Advanced Simulation Technologies Conference (ASTC), April 22-26, 2001, Seattle Washington, Manuscript Number M172.
21. C. A. Dreyer, D. L. Shoch, *'F-16 Flutter Testing at Eglin Air Force Base,'* AIAA Paper 86-9819, April 1986.
22. J. D. Anderson, *'Introduction to Flight,'* McGraw-Hill, Inc. New York, New York, 1989, pp. 120-121.
23. E. H. Dowell, J. P. Thomas, K. C. Hall, *'Transonic Limit Cycle Oscillation Analysis Using Reduced Order Aerodynamic Models,'* 42<sup>nd</sup> AIAA/ASME/ASCE/AHS/AHC Structures, Structural Dynamics, and Materials Conference, 16-19 April 2001, Seattle, Washington.
24. W. A. Silva, *Discrete-Time Linear and Nonlinear Aerodynamic Impulse Responses for Efficient CFD Analysis,* Ph.D. Dissertation, The College of William and Mary in Virginia, Williamsburg, Virginia, 1997.

25. R. P. Prazenica, A. Kurdila, W. Silva, *Multiresolution Methods for Representation of Volterra Series and Dynamical Systems*, 41<sup>st</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, Atlanta GA, 3-4 April 2000.
26. D. E. Raveh, *Aircraft Aeroelastic Analysis and Design using CFD-Based Unsteady Loads*, 41<sup>st</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, Atlanta GA, 3-4 April 2000.
27. D. E. Raveh, D. N. Mavis, *Reduced-Order Models Based on CFD Impulse and Step Responses*, 42<sup>nd</sup> AIAA/ASME/ASCE/AHS/AHC Structures, Structural Dynamics, and Materials Conference, 16-19 April 2001, Seattle, Washington.
28. A. M. Cunningham, Jr., *The Role of Non-Linear Aerodynamics In Fluid-Structure Interaction*, 29<sup>th</sup> AIAA Fluid Dynamics Conference, Albuquerque, NM, June 15-18 1998.
29. A. Sedaghat, J. E. Cooper, J. R. Wright, and A. Y. T. Leung, *Limit Cycle Oscillation Prediction for Aeroelastic Systems with Continuous Non-Linearities*, 41<sup>st</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, Atlanta GA, 3-4 April 2000.
30. R. Lind, *'Flight Testing with the Flutterometer,'* 42<sup>nd</sup> AIAA/ASME/ASCE/AHS/AHC Structures, Structural Dynamics, and Materials Conference, 16-19 April 2001, Seattle, Washington.
31. S. Potter, R. Lind, *'Developing Uncertainty Models for Robust Flutter Analysis using Ground Vibration Test Data,'* 42<sup>nd</sup> AIAA/ASME/ASCE/AHS/AHC Structures, Structural Dynamics, and Materials Conference, 16-19 April 2001, Seattle, Washington.
32. W. E. Faller, S. J. Schreck, and M. W. Luttges, *'Neural Network Prediction and Control of Three-Dimensional Unsteady Separated Flowfields,'* AIAA Journal of Aircraft, Vol. 32 No. 6, November-December 1995.
33. W. E. Faller, S. J. Schreck, and H. E. Helin, *'Real-Time Model of Three-Dimensional Dynamic Reattachment Using Neural Networks,'* AIAA Journal of Aircraft, Vol. 32, No. 6, November-December 1995.
34. J. E. Cooper, W. J. Crowther, *'Flutter Speed Prediction During Flight Testing Using Neural Networks,'* CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, Va, June 1999.
35. Y. S. Wong, B. H. K. Lee, and T. K. S. Wong, *'Parameter Estimation in Flutter Analysis by Wavelet and Neural Network,'* CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, Va, June 1999.

36. R. L. McMillen, J. E. Steck, K. Rokhsaz, "Application of an Artificial Neural Network as a Flight Test Data Estimator," AIAA Journal of Aircraft, Vol. 32, No. 5, September-October 1995.
37. L. L. M. Pucci and M. Pecora, "Flutter Speed Prediction Using the Artificial Neural Network Approach," Department of Aeronautical Engineering, University of Naples "Federico II", Naples, Italy, Source and date Unknown.
38. P. F. Lichtenwalner, G. R. Little, and L. E. Pado, "Adaptive Neural Control for Active Flutter Suppression," FED-Vol. 242, Proceedings of the ASME Fluids Engineering Division, ASME 1996.
39. H. J. Cunningham, J. T. Batina, R. M. Bennett, "Modern Wing Flutter Analysis by Computational Fluid Dynamics Method," Journal of Aircraft, Vol. 25, No. 10, October 1988, pp. 962-968.
40. J. P. Thomas, E. H. Dowell, K. C. Hall, "Nonlinear Inviscid Aerodynamic Effects on Transonic Divergence, Flutter and Limit Cycle Oscillations," 42<sup>nd</sup> AIAA/ASME/ASCE/AHS/AHC Structures, Structural Dynamics, and Materials Conference, 16-19 April 2001, Seattle, Washington.
41. L. Tang, R. E. Bartels, P. C. Chen, D. D. Liu, "Simulation of Transonic Limit Cycle Oscillations Using a CFD Time-Marching Method," 42<sup>nd</sup> AIAA/ASME/ASCE/AHS/AHC Structures, Structural Dynamics, and Materials Conference, 16-19 April 2001, Seattle, Washington.
42. J. Kohlmorgen, K. R. Muller, K. Pawelzik, "Analysis of Drifting Dynamics with Neural Network Hidden Markov Models," in Jordan M.I., et al., Advances in Neural Information Processing Systems 10, MIT Press/Bradford Books, Cambridge/London, 1998, pp.735-741.
43. D. Wang, "Emergent Synchrony in Locally Coupled Neural Oscillators," IEEE Transactions on Neural Networks, Vol. 6, No. 4, July 1995.
44. H. Kurokawa, C. Y. Ho, S. Mori, "The Stability of the Synchronization Learning of the Oscillatory Neural Networks," 1997 IEEE International Symposium on Circuits and Systems, June 9-12, 1997, Hong Kong, pp. 513-516.
45. J. Kuo, "Nonlinear Dynamic Modeling with Artificial Neural Networks," Ph.D. Dissertation, University of Florida, 1993, pp. 126-134.
46. J. C. Principe, N. R. Euliano, W. C. Lefebvre, "Neural and Adaptive Systems: Fundamentals through Simulations," John Wiley & Sons, Inc., New York, New York, 2000, pp. 558-564.
47. D. Wang, "Emergent Synchrony in Locally Coupled Neural Oscillators," IEEE Transactions on Neural Networks, Vol. 6, No. 4, July 1995.

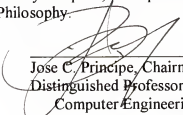
48. R. D. Strum, D. E. Kirk, '*First Principles of Discrete Systems and Digital Signal Processing*,' Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1988.
49. T. Kailath, "*Linear Systems*," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1980.
50. G. E. P. Box, G. M. Jenkins, G. C. Reinsel, G. Jenkins, "*Time Series Analysis: Forecasting and Control*," Prentice Hall, Upper Saddle River, New Jersey, 1994.
51. S. Haykin, J. C. Principe, '*Dynamic Modeling of Chaotic Time Series with Neural Networks*,' IEEE DSP Magazine, May 1998, pp. 66-81.
52. J. C. Principe, L. Wang, M. A. Motter, "*Local dynamic modeling with self-organizing maps and applications to nonlinear system identification and control*", Proceedings of IEEE, Vol. 86, No. 11, Nov. 1998, pp. 2240-2258.
53. J. A. K. Suykens, J. Vandewalle, (editors), '*Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling*,' Katholieke Universiteit, Leuven, Belgium, July 1998.
54. F. Takens, '*Detecting Strange Attractors in Turbulence*,' Dynamical Systems and Turbulence, D. A. Rand, L. S. Young (editors), Springer Lecture Notes in Mathematics, Springer-Verlag, New York, 1980, pp. 365-381.
55. K. S. Narendra, '*Neural Networks for Intelligent Control*,' 1997 American Control Conference, Workshop Number 4, Albuquerque, New Mexico, and The Center for Systems Science, Yale University, 1997, p. 141.
56. U. Appel, A. V. Brandt, '*Adaptive Sequential Segmentation of Piecewise Stationary Time Series*,' Inf. Sci., vol. 29, no.1, 1982, pp. 27-56.
57. M. Basseville, I. V. Nikiforov, '*Detection of Abrupt Changes, Theory and Application*,' Prentice-Hall, Englewood Cliffs, New Jersey, 1993.
58. M. Riedmiller, H. Braun, "*A direct adaptive method for faster backpropagation learning: The RPROP algorithm*," Proceedings of the IEEE International Conference on Neural Networks, 1993.
59. J. C. Principe, N R. Euliano, W. C. Lefebvre, '*Neural and Adaptive Systems, Fundamentals of Through Simulations*,' John Wiley and Sons, Inc., 2000, p. 208.
60. C. W. Helstrom, '*Probability and Stochastic Processes for Engineers*,' Macmillan Publishing Company, New York, New York, 1984, p. 254.



## BIOGRAPHICAL SKETCH


Mike Johnson was born in 1958. He attended the University of Utah and earned a Bachelor of Science degree in electrical engineering in 1986. He started working for the United States Air Force at Hill Air Force Base that same year as an electronics engineer and in 1987 transferred to Eglin Air Force Base in Florida. In 1998 he earned a Master of Engineering degree in electrical and computer engineering from the University of Florida. During his career he became interested in the flutter and limit cycle oscillation phenomenon as a signal processing problem, leading to a doctoral research topic. He continues research in the field as a square-peg electrical engineer in a round-hole world of aeroelasticians.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.




\_\_\_\_\_  
Jose C. Principe, Chairman  
Distinguished Professor of Electrical and  
Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



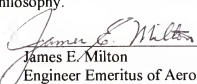
\_\_\_\_\_  
Henry Zmuda  
Associate Professor of Electrical and  
Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



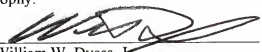
\_\_\_\_\_  
Tan F. Wong  
Assistant Professor of Electrical and  
Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



\_\_\_\_\_  
James E. Milton  
Engineer Emeritus of Aerospace  
Engineering, Mechanics and  
Engineering Sciences

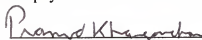
I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



\_\_\_\_\_  
William W. Dyess, Jr.  
Deputy Director, Air Armament Material  
Wing, Eglin Air Force Base, Florida

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 2003

A handwritten signature in dark ink, appearing to read "Pramod P. Khargonekar", is written over a horizontal line.

Pramod P. Khargonekar  
Dean, College of Engineering

---

Winfred M. Phillips  
Dean, Graduate School